A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

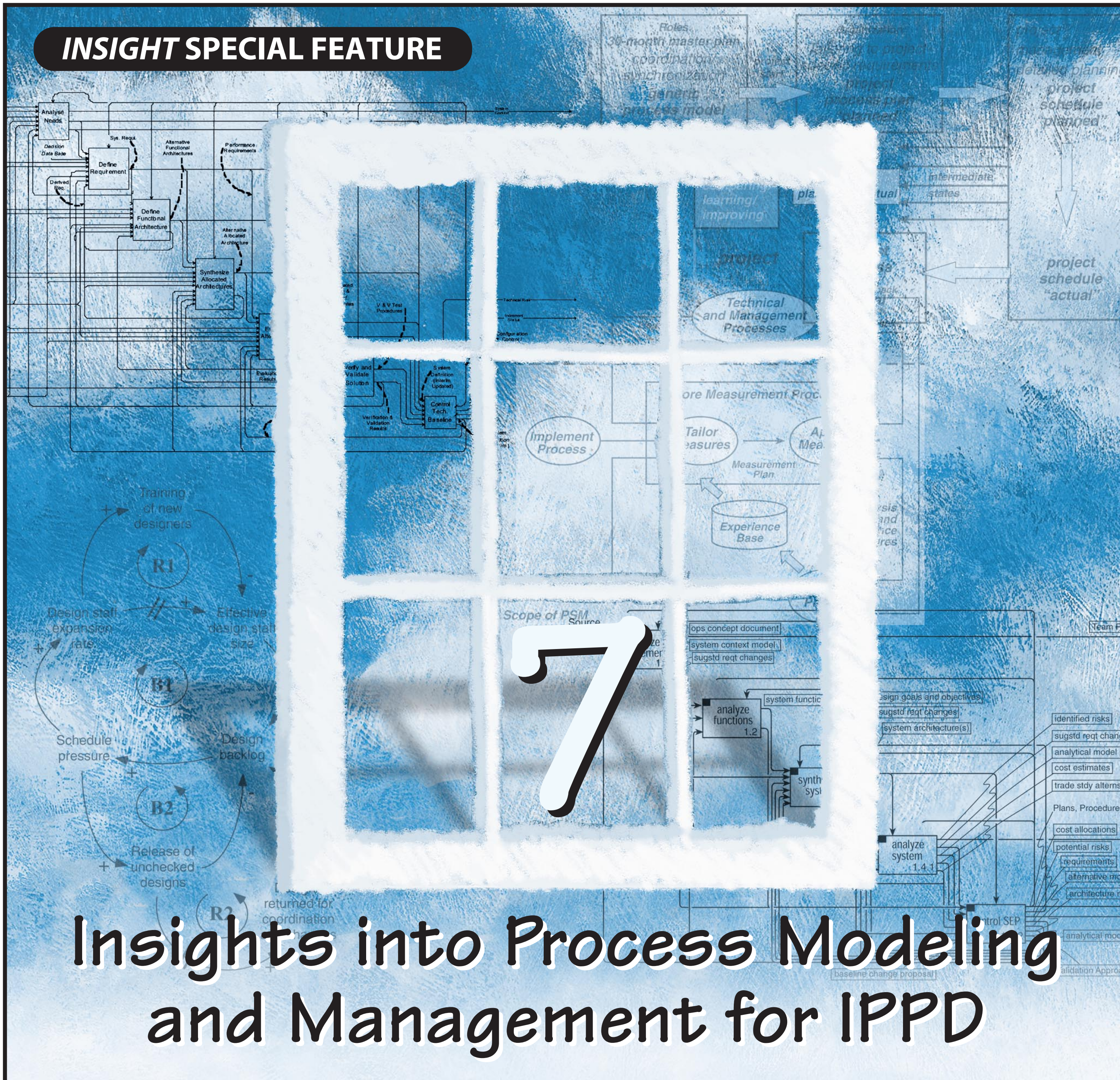
Fall 1999

INCOSE

Vol 2 Issue 3

INSIGHT

INSIGHT SPECIAL FEATURE



7

Insights into Process Modeling and Management for IPPD

INCOSE
INSIGHT
A PUBLICATION OF THE INTERNATIONAL COUNCIL OF SYSTEMS ENGINEERING

Managing Editor Valerie Gundrum
valerie.gundrum@lmco.com (607) 751-2245
Theme Editors Ernst Fricke, Herbert Negele
e.fricke@lrt.mw.tum.de
h.negele@lrt.mw.tum.de
Advertisement Editor Bill Wittig
bwittig@usa.net (517) 757-3057
Production Editor Shirley Bishop
incose@halcyon.com (800) 366-1164
Layout and Design Chuck Eng
incose@chuckeng-design.com (206) 546-9698
Member Services INCOSE Central Office
incose@halcyon.com (800) 366-1164
Web Services Lew Lee, Webmaster
lew.lee@trw.com (408) 743-6474

On the WWW http://www.incose.org

Article Submission INSIGHT@incose.org

Publication Schedule. ***INSIGHT*** is published four times per year. Issue and article/advertisement submission deadlines are as follows: Winter 1999—November 23, Spring 2000—February 15, Summer 2000—May 16, Fall 2000 – August 15. For further information on submissions and issue themes, visit the INCOSE website as listed above.

Subscriptions to *INSIGHT* are available to INCOSE members as part of their membership. Complimentary copies are available on a limited basis. To inquire about membership or a copy, contact Member Services.

Copy Editors: Lori Pajerek, Shirley Bishop

From the Editors

Seven Insights into Process Modeling and Management for IPPD

It seems to be a tradition that almost every conference or workshop on systems engineering or product development be dominated by "faster, better, and cheaper" concepts. In order to make this slogan come true, most companies try to reengineer and improve their product development activities by defining and implementing new development processes. The testimonies at the conferences show that the typical goal of such an endeavor is to reduce development times by fifty percent and development costs at least by a quarter.

In their activities to set up a new development process, systems engineers and systems engineering principles should play an crucial role, since systems engineering is the "glue" that brings together the activities of the different disciplines, and ensures a complete systems view. As we know, system development is comprised not only of the product system, but also the process system, and therefore system engineers also need a comprehensive view of the entire development process. That is why we think "process modeling and management" is an important topic for INCOSE, the systems engineering community, and everybody involved in product development.

Usually, in the systems engineering community, many immediately relate process modeling, management and improvement to CMM initiatives. But this theme issue wants to highlight a different aspect of this topic: How to realistically analyze and model your engineering processes in enough detail to really understand and improve them.

Typically after reengineering activities or in big new projects, employees no longer know "where they are" in the development process, with whom they have to communicate, and what is expected from them. Here, organization-level process modeling and management could be of great help. We have also found that many people advocate model-based system development using state-of-the-art techniques. But, before you buy and implement sophisticated, expensive tools to support these methods, you first have to understand your process and its information links in order to understand how it can be improved by using all the tools.

Product development and engineering design processes differ from other business processes such as logistics, supply chain, or manufacturing processes, where process modeling is widely used. It's the special characteristics (like creative, fuzzy, interdisciplinary, iterative...) that make modeling of development processes more challenging, especially in an integrated product and process development (IPPD), or concurrent engineering, environment. This has to be regarded when talking about describing and improving development processes.

This theme issue is comprised of "seven insights," or seven articles, into process modeling and management. Obviously, these seven viewpoints cannot cover the whole topic exhaustively, but should provide INCOSE members with an overview of the whys, whats and hows of process modeling and management in an IPPD environment. We were fortunate to get authors from industry and academia representing four different countries. First, as the theme editors for this issue, Herbert Negele

continued on page 5

President's Corner

Ken Ptack, ptack_ken@prc.com

I was enthusiastic about my plans to help John Clouet and the Silver State Chapter open the 13th International Conference on Systems Engineering (ICSE) "The Many Faces of Systems Engineering" in Las Vegas, Nevada 9–12 August 1999. On the Friday before, I went to the hospital for what I assumed would be a normal stress test. Unfortunately, I ran into one of those many faces of systems engineering, and experienced a "system fault" that changed my plans. I was four minutes into the stress test when the doctor went "all stop," put me in a wheelchair, and admitted me to the cardiac unit. I experienced new stress that weekend — they kept me in bed and diligently monitored my heart. On Monday, the first day of the Las Vegas event, my time was spent conferring with the doctors and other medical specialists, rather than giving the keynote address. The diagnosis was that one of the arteries feeding my heart was 90-95% blocked. On Tuesday, I had emergency cardiac surgery in which they inserted a coronary stent implant by way of an angioplasty. I was released from the hospital on Wednesday, with the restriction that I could not travel for a few weeks. Thanks to modern systems and technology, the doctor was able to determine that I had a problem before my body told me I did. My body was operating in a degraded mode, which fortunately did not succumb to unexpected total system failure. Fortunately, everything now appears to be "all systems go!"

Having read the above, you ask yourself what does this have to do with systems engineering? It illustrates that there are many systems that depend on other systems to tell you when and if something is wrong. It tells you that systems include people, machines, software, interfaces, and an overall understanding of how the system should work. In addition, it shows us how much we depend on others to help us understand and find an acceptable solution to our problems.

Since the last issue of ***INSIGHT*** (and in spite of my medical setback!), INCOSE continues moving towards its goals — the future is exciting. In the remainder of this article, I would like to provide highlights to several activities: the conference in Las Vegas, changes in leadership positions, technical liaisons, and upcoming events.

The 13th ICSE, mentioned above, was co-hosted by the Silver State Chapter and the University of Nevada, Las Vegas. They had an impressive turnout and program, with nearly 300 attendees, half of which were INCOSE members. Eric Honour did yeoman's duty and was able to fill in for me. In part, he had the following to say about the conference:

"I was thoroughly pleased with the quality and variety of papers presented, with the quality and variety of speakers, with the level of support from the hotel and the conference organizers, with the condition of the spaces used, and much more. This conference speaks very highly of UNLV, TRW and the other sponsors, the ICSE series, and of INCOSE. I would like to thank you and all your associates for an excellent conference."

I would like to highlight other important accomplishments brought about by the diligence of our members. INCOSE continues to forge international partnerships, namely with AFIS and ISO. We have recently completed a Memorandum of Understanding with the Association Française d'Ingénierie Système France (AFIS). AFIS will represent and promote the INCOSE mission in France, and AFIS individual members will be members of INCOSE. At the International Standards level, INCOSE has striven to establish a Category "A" Liaison with JTC1/SC7. The SC7 working group has approved INCOSE's request and it is anticipated that JTC1 will also approve it, as will ISO Central. This will give INCOSE a direct advisory role in JTC1/SC7, and in the development of Systems Engineering related standards, as opposed to an INCOSE member only "being recognized" by a Country's Technical Advisory Group (TAG). In addition, we have provided comments on the current draft of ISO 15288.

INCOSE is sponsoring several upcoming events. Each year's symposium is a premier event, and INCOSE 2000, July 16-20 in Minneapolis, will be a grand event. (Mark your calendars now!) In addition, we have Regional Conferences planned for Europe, Australia, and the United States (you will find more information on the INCOSE web site and in this newsletter). These events provide more opportunities to participate in INCOSE activities, and to learn and share systems engineering experiences. In addition, Donna Rhodes is leading the charge on the INCOSE Strategy 2000, maintaining a broad perspective that will set the vector for the future of INCOSE.

There have been several changes to chair positions in the organization. Terry Creque has accepted the position of Ways & Means Co-chair, and will be working closely with Bob Kenley to maintain procedural structure and order, and ensure that we are operating in accordance with INCOSE Policy. Cecilia Haskins has accepted the Co-Chair position of the Chapters Committee, working with Sam Rindskopf to enhance the products and services offered to new and existing chapters. Lastly, Cassandra Fleetwood has accepted the position of Co-chair of the Communications Committee, and Valerie Gundrum has stepped up to chair this group. My warmest regards to the "retiring" chairs of these three committees: Joe DeFoe, Ken Kepchar, and Randy Case. You helped lay the foundation for your successors.

All in all, I see INCOSE continuing to grow, and to gain recognition and respect from all over the world.

We have come a long way, but still have far to go. It is up to all of us to continue our success and growth. Please tell a co-worker, friend and neighbor about the benefits that you gain from your membership and encourage them to join in the fun.

Kenneth R Ptack

Ken Ptack
INCOSE President
Litton/PRS Inc.

Cancelled!

Seattle Metropolitan Chapter
Mini-Conference
Re-scheduled for Fall 2000

We regret any inconvenience.

For more information, contact Herman Migliore at
hermm@cs.pdx.edu

From the Editors *continued from page 3*

and Ernst Fricke, Technical University of Munich, with Nicole Härtlein, BMW AG, describe the need for modeling engineering processes, its benefits and constraints, as well as their view of a suitable method. Our article gives a discussion of the range of process modeling and management for product development processes and shall guide the reader into the topic.

Jack Ring, Kennen Inc., sets out his position that a model communicates a vision, and if all we model is process, then all we will get is process. Therefore, a process-only view is not sufficient, but a project model is required that also includes market and enterprise development activities, information flows, resources and other aspects. From his example he derives requirements for a suitable modeling language. This constitutes a good basis to get a more detailed view on specific approaches to modeling and managing processes, following different goals.

Tyson Browning, Lockheed Martin Corporation, shows us why and how to use the Design Structure Matrix (DSM). This powerful method allows one to visualize, analyze, and improve product development processes effectively, helping people to see, after a quick orientation, how their activities affect a large process. A DSM clearly describes the dependencies among process elements and conveniently highlights iteration and rework. Tyson also presents us with an example of how the DSM is applied in real-life.

We get our next insight from Rick Steiner and Doug Stemm, Raytheon Systems Company, into an ongoing project to develop and deploy a Systems Engineering Management Plan (SEMP) model, using an SE tool and the IDEF0 method. They explain how it has helped to focus the systems engineering process across the widely distributed project team. The SEMP model is intended to be used as a proper vehicle for process understanding and innovation.

David Ford, University of Bergen, describes an application of system dynamics to concurrent engineering for managing and improving processes. He demonstrates the value of system dynamics as a systems engineering tool to investigate the interactions of product development processes, resources, management and participant behavior at the operational level.

From Len Karas and Donna Rhodes, Lockheed Martin Federal Systems (LMFS), we learn about the latest enhancement to the systems engineering methodology in use at LMFS. They use an Operational Description Template (ODT) to enable a multi-discipline team to effectively understand and specify how the system will work. Instead of explicitly modeling the development process, ODTs become the basis for the entire team's work products and the common denominator within the product development process. Additionally, they have begun to use the ODT for modeling process and planned technology infrastructure improvements.

Finally, Claude Laporte, Yortar Technologies, and Sylvie Trudel, Oerlikon Aerospace, tell us about a systems engineering process improvement initiative at Oerlikon Aerospace. A brief description of the context is given, then the systems engineering process and organizational mechanisms to better manage changes are described. Finally, lessons learned are presented which highlight that in all process modeling and improvement initiatives, we constantly should pay attention to the "people issues."

We want to thank all authors for their contributions to this theme issue, hoping it is a valuable contribution for all INCOSE members. Comments are greatly welcomed!

Best regards,
Ernst Fricke & Herbert Negele
Theme Editors

INSIGHT SPECIAL FEATURE

Modeling Concurrent Engineering Processes in an Integrated Product Development Environment

Herbert Negele, h.negele@lrt.mw.tum.de, Ernst Fricke, e.fricke@lrt.mw.tum.de, Nicole Härtlein, nicole.haertlein@bmw.de

Introduction. In order to successfully produce today's highly complex and integrated systems, many companies have reengineered their product design and development activities, since quality and costs of a product are determined mainly in this stage of the life cycle. New management concepts (e.g., Total Quality Management, Concurrent Engineering) and organizational concepts (e.g., team-oriented organizations with Integrated Product Teams) also strongly impacted the way of developing and increased the amount of information that has to be dealt with and exchanged in the development phase. Globally distributed development sites and close cooperation with suppliers from all over the world lead to a "scattered" development process with lots of interfaces that, nevertheless, have to be coordinated effectively. These and other changes have altered the method of developing products significantly. Therefore, in today's companies there is a strong need for means that enable better understanding, documentation, communication, and learning, especially with regard to development processes and the inherent process know-how.

For a comprehensive view of a development system, besides the processes, additional (interrelated) aspects have to be taken into account (e.g., market, customer and user needs, requirements and goals, products, people, resources, organizations). Systemic methods and tools can help manage these complex systems successfully by enabling a "systems view" modeling and analysis of all relevant elements and their interrelations. One method that we have proposed is the "ZOPH Model," a comprehensive systems modeling approach embracing, structuring, modeling, and interrelating information essential for product development systems (Negele et al. 1997, Negele 1998). It structures all the information relevant to a given development system by using the following different system types

- system environment
- development system (project system)
  - **Z**ielsystem (goal system),
  - **O**bjektsystem (product system),
  - **P**rozeßsystem (process system), and
  - **H**andlungssystem (agent system).

As part of this ***INSIGHT's*** theme, we want to focus on the development processes (process system) and how they can be modeled in order to meet the requirements arising from their specific characteristics, taking into account a concurrent engineering environment with de-centrally acting teams and individuals.

Why Model Development Processes? Several arguments can be brought forward to answer this question (Fricke et al. 1998):

- *Transparency:* A process model helps people to get an overview, to perceive what part they play in the game, and to understand who is doing what, why and when. This is all the more necessary when standard processes are changing (e.g., due to reengineering).
- *Understanding and Learning:* A transparent process model supports and communicates understanding of complex processes and their interactions and dependencies within the organization. It also provides an excellent learning aid for employees who are new or have changed jobs.
- *Coordination:* In the course of the development, many process interfaces (especially information flows) have to be coordinated. Increasingly, this has to be accomplished across globally distributed organizations. A consistent process model promotes better communication (people talk about the same things) and allows early planning of future actions and interactions.
- *Better planning and management:* By enabling transparency and early coordination, a modeled process represents a sound basis for detailed planning and easier management of the actual development project.
- *Documentation and Reusability:* Process models capture process know-how and are a kind of documentation that can be fully or partially reused as a starting point or "building blocks" in subsequent development projects.
- *Prerequisites for Audits:* Achieving certification (e.g., ISO 9001) requires a documented process and evidence that the process is performed as documented. A process model that is used (or "lived") by all people involved in the development activity can provide both.
- *What-if Analyses:* A process model can be used to conduct what-if analyses to determine the effects of process changes. Moreover, process simulation capabili-

ties can be built upon the model.
- *Basis for Process Assessment and Improvement:* Only if you know what you are doing (which can be described in a process model), can you assess how well you are doing it and use it as a basis for improvement.
- *Shorter Development Cycles:* One main reason for process modeling is to achieve shorter development times. Process models can be the basis for process reengineering and optimization activities.

In order to realize the benefits resulting from creating and using a process model, a better understanding of the specific characteristics of these processes is necessary.

Characteristics of Development Processes. Many development activities have a unique and intuitive character that is very difficult to capture in a model. Decisions have to be made early and rely on assumptions, such as future market trends or availability of innovative technologies. Because of the newness of many development tasks, acquired information often is tainted with uncertainty. Usually, development processes are treated as what we call "sequential" processes. Today's development processes differ significantly from other business processes, like production, logistics, or supply-chain processes. While these resemble sequential process chains that are performed several times in a very similar or identical way, development processes are more like *process nets* (see above), with processes that are dynamically changing and highly interconnected, including feedback-loops and interactions on different levels.

A detailed analysis of tools for modeling business processes revealed that none of the available tools met the requirements derived for modeling concurrent engineering processes for integrated systems (Fricke et al. 1998). There seems to be a lack of understanding of the special characteristics of development processes. Methods and tools developed for modeling and analyzing "sequential" processes were simply applied to development processes. This might also be a reason for the failure of many reengineering activities in the product development arena. Also, many modeling methods and tools were developed for and used in projects whose focus was the reengineering of business processes in preparation for the introduction of new IT systems and were therefore driven by the needs of IT specialists (Lullies et al. 1998).

Some Fundamental Aspects. In general, two different approaches to process modeling can be distinguished: de-central or central. This has a strong impact on factors like the type and number of process modelers, consistency and "density" of the information modeled (Fricke et al. 1998). With the central approach, typically, some process specialists collect all relevant information, then analyze, structure, and model it top-down. An advantage of this approach is that modelers can be specially trained for their jobs, along with the supporting methods and tools. Since process modeling is their main task, they are capable of generating sophis ticated models with a high "information density." As the number of modelers here normally is quite small, the modeled information should be quite consistent in regard to the content and degree of detail.

A fundamentally different approach is to let everyone involved in product development work on the process model de-centrally. The number of modelers or users of a corresponding process modeling tool can be several hundred (e.g., in automotive development), where dozens of people may work on the model concurrently. An advantage of this approach is that all people are using the model on a regular basis, the information contained need not be collected by the specialists first and therefore should be quite up-to-date. Since the modelers (process owners) know their processes best and are not (mis)interpreted by others, the model is likely to be more realistic. Also, less effort should be required to update the process model because many individuals share it, and forwarding or even loss of information can be avoided.

Not surprisingly, the best alternative was found in a combination of both approaches. At the project partner's site, several unsuccessful attempts had been made to build up a detailed central process model. Therefore, a combined top-down and bottom-up approach was chosen. In order to provide a guiding framework, a centrally generated and coordinated master plan and a common, top-down process model structure were used for the practical integration of the distributed, bottom-up modeling efforts.

For a reengineering project, engineers had started to model their processes with a quite simple input-process-output (IPO) logic, describing what they are doing (P), what they need to do it (I), and what they produce (O). The output of one process can be used as input by other processes. These output-input relations represent the interactions and flows between processes. For the IPO descriptions, ordinary MS Word forms were used that everyone could generate. But there was no way to support output-input links between processes described on different IPO forms.

Besides the IPO forms, in daily practice many different methods and tools for process modeling were used (CAD-tools, spreadsheets, project scheduling tools, etc.). All the various process descriptions created in that manner were used by the engineers to plan their future activities. Due to different formats, the generated data "islands" could not be linked or exchanged, resulting in increased planning and coordination efforts. Therefore, the need for a common tool supporting modeling, planning, and coordination of processes and their interactions became obvious.

For such an endeavor, several boundary conditions have to be taken into account. Since the workload of the engineers generally is very high, it is crucial they have an operational need and benefit of applying such a tool. It has to be easy and intuitive to use, as modeling processes is not normally the main task of the engineers. Moreover, engineers have to overcome the conviction that everything they are doing is unique and therefore can't be modeled. Fricke (1998) describes principles and methods for realizing such a user-centered approach.

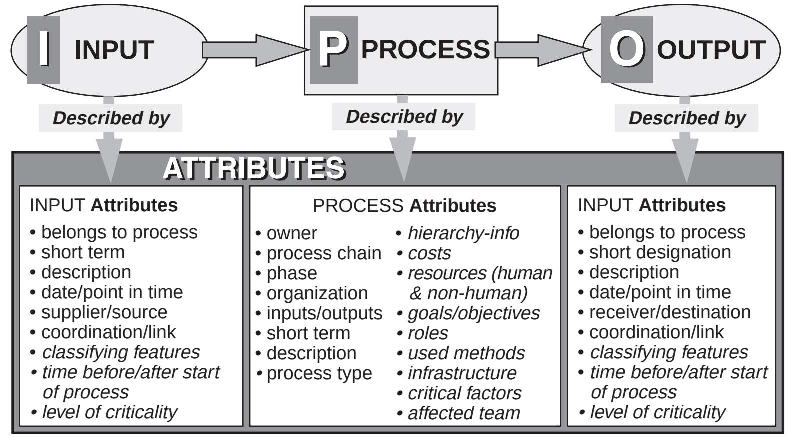


Figure 1. Attributes describing the IPO-elements

Description of the IPO Method.

Basic Components of the IPO-Method. Following the approach already used in the reengineering efforts mentioned above, and in order to be able to reuse the information already collected, the following basic components of the process model were defined (Negele et al. 1999):

- *Processes:* describe relevant tasks and activities of the people involved; events (e.g. milestones) are seen as special cases of processes (no temporal extension)
- *Inputs:* represent input objects necessary to carry out the process, e.g. documents, data files, software or hardware models
- *Outputs:* represent objects that are produced or worked on in the process, e.g. documents, data files, software or hardware models
- *Links:* describe interactions between processes (flow of information, matter, etc.) and define output-input interfaces.

A process with its assigned inputs and outputs can be understood as the fundamental building block ("IPO-element") for the process model. Building blocks are interlinked by connecting outputs and inputs of (usually different) processes. A single output can be linked to several inputs (e.g., a requirements document is needed in several processes).

The basic components of the process modeling language can be described in more detail by many different attributes (Figure 1). For example, information on costs, risks, confidence, resources, applied methods and tools, relevant objectives/requirements, type of process, etc. can be assigned to the processes, besides normal information like title, description, owner, and duration. To be prepared for changing (user) requirements, the set of attributes can be changed or extended flexibly at any time.

Linking Concept. For interlinking processes, temporal dependencies that are often used (e.g., in network planning) are not sufficient. There is a need for "meaningful" relations representing flows (especially of information and material) between processes and interactions between process owners, i.e., individuals, teams, or other organizational entities. Therefore, the output-input links used in the IPO method enable involved process owners (who) to interactively make agreements on content (what) and time (when) of their interactions. Inputs and outputs can be assigned to any point in time within the process duration. Additional information on problems/objectives (why), locations (where), means (how), coordination status, etc. can be assigned to the processes/relations according to specific needs by defining corresponding attributes. Moreover, different types of relations can be distinguished, for example with regard to the duration, the direction of an interaction, type of exchanged data, importance, or criticality.

With this linking concept, an effective, de-central interface management and process coordination within and across projects can be supported. Direct communication between the involved persons and teams is not to be replaced by establishing such a process model. Rather, the IPO-Method can help to easily determine where and when interaction and communication is necessary, and assist in planning and coordination activities.

Structuring the Process Model. A common framework for the process model has to be set up to assist the process owners in integrating their processes into the whole picture. In order to achieve this, the process net can be structured in several dimensions. The three main dimensions are process chain (e.g. chassis, engine, etc.), organizational entity or role, and development phase (defined in the master plan).

Hierarchical concept. Most methods (e.g., SADT/IDEF) use a strict hierarchical, top-down approach to model processes where interrelations between processes can be described on the same level of detail only. In practice, there are manifold logical, informational, temporal, and other interactions between the different process steps across all levels of detail. A strict hierarchy (similar to a strictly functional organization) is not adequate for process modeling in today's product development environment (relying on team-oriented, cross functional and/or virtual structures). On the other hand, the total renouncement of a hierarchy will result in a very confusing process model. Therefore, the concept of a "pseudo-hierarchy" was developed (Negele et al. 1999). In this hierarchical concept the structuring criterion is put into an attribute. The instances of the attribute, e.g. the process chains, are hierarchically ordered. This structure is valid for the entire process net, and is not dependent on duration or other factors. Every single process step is then hooked up to an instance of this pseudo-hierarchy, thus enabling the clustering and filtering of process steps that belong to the same instance (e.g. the process chain "geometrical integration"). Several pseudo-hierarchies for different attributes can be built up and combined. This results in an ordered process map, where all processes can be linked directly across all hierarchies, supporting a de-central and agile modeling approach.

Process Model Life Cycle. Ideally, the life cycle of the process model should look like it is shown in Figure 2. A single use of a process model would be inefficient. Therefore, a generic process model, which is based on the master plan, should be the bedrock for all projects. At each project start, the generic process model is tailored to project-specific requirements, defining the project process plan. While running the project, the processes will be modeled in more detail and used for the project schedule. Because there is no innovative process without changes, the project schedule will be subject to on-going changes. At the end of the project, the actual (as-done) project schedule can be used as an actual process plan to compare planned vs. actual process plan. Certainly, this can also be done with intermediate states of the project schedule. This helps people to learn from each project and continuously improve the generic process model. For practical reasons (less effort) the first generic process model can also be generated from the last well-run project and its projects schedule.

Integration of process view and schedule view. The process model and the scheduling model are based on the same modeling components in order to transfer easily the planned process map into the scheduling tool (and vice-versa) when running a project. This provides the users with an operational benefit, as they can use the process model later in their own projects. Additionally this supports the possibility of taking a process view of an already running project based on the data from the scheduling tool. Essentially, these are just two different views of the same data.

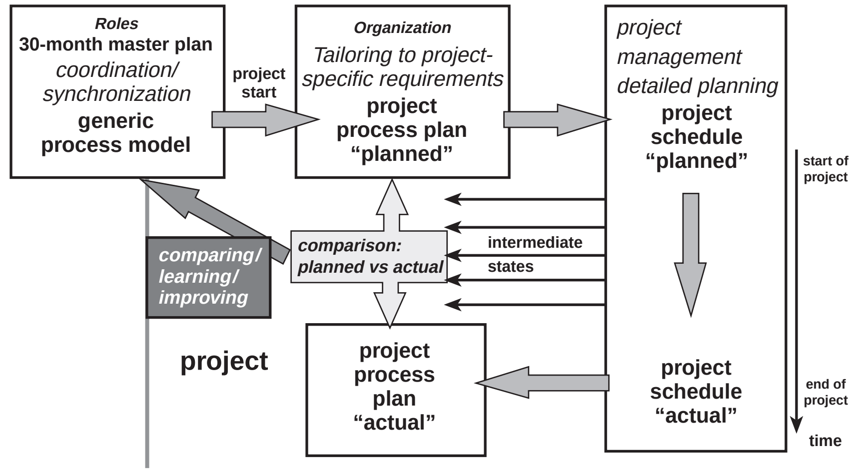


Figure 2. Process Model Life Cycle

Application of the IPO-Method. To support the IPO method, a specification for a SW-tool (TIPO) and an operational concept were generated (see Figure 3). A prototype was implemented and already applied to modeling parts of the development processes. For example, if Mr. Franz needs a "preliminary concept" for his process "validate concept" from Mr. Hans, then he will create that input he is expecting from him in the TIPO tool. This input is now put into a list of required inputs. It will be labeled by the status "not coordinated," visualized by a red traffic light. At some point in time, Mr. Hans can check with a filter in list to determine whether there are inputs requested from him, meaning he has to deliver an output. When he finds an input in the list that Mr. Franz needs from him, he can connect (using drag and drop) this requested input to his process, thus generating the corresponding output. The status of the created link is "in coordination," shown by a yellow traffic light. Then Mr. Hans can either accept based on the attributes of this output (e.g., contents and point in time of delivery) or make a change proposal to Mr. Franz. When they finally agree, the status of the input/output link is changed to "coordinated" (green traffic light). This coordination process and a graphical visualization (boxes and arrows) of the process net help both partners to have the same understanding of what will be delivered and when. Also, it supports an analysis of the process net, to find out which processes are still not coordinated, who generates outputs nobody wants as an input, or where necessary inputs are requested but nobody agreed to provide them. So, this method helps to understand critical issues early in process planning, and not at a point in the project when someone is desperately waiting for a delivery.

Conclusion. One of the main reasons that engineering processes have to be modeled and documented is the ongoing efforts in all industries to reengineer their development processes. Integrated product development differs significantly from other, sequential business processes. It rather resembles a process net, including many-fold interrelations, feedback-loops and interaction on different hierarchy levels. The focus

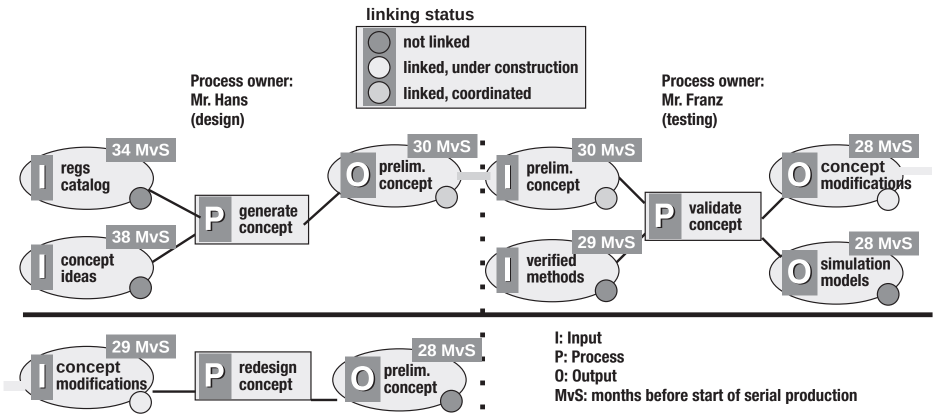


Figure 3: Coordination Concept

for modeling integrated product development processes has to be on the interfaces, i.e. the information flow, shifting the view to an information-based process model.

Two different approaches can be distinguished for process modeling: de-central (everyone) and central (specialists). The advantage of a mixed bottom-up and top-down, but strongly de-central, approach is that it enables having a "living," up-to-date process. To make it work in daily practice, the simple, but powerful IPO process modeling method and a flexible hierarchical concept was chosen. All engineers have to have an operational benefit from modeling their processes, which is supported by using the modeled processes later on in their project scheduling, as well.

Further work has to prove the benefit of the presented approach in a company-wide daily practice. Up to now, the method itself was highly accepted and the presented approach is used in several small projects. The SW-tool TIPO has to be improved with regard to functionality and ease of use. Also, process metrics allowing for a tangible analysis of the IPO process models have to be developed in order to facilitate process improvement.

References

Fricke, E.: Der Änderungsprozeß als Grundlage einer nutzerzentrierten Systementwicklung. Ph.D. thesis. Technische Universität München, Herbert Utz Verlag Wissenschaft, 1999.

Fricke, E.; Negele, H.; Schrepfer, L.; Dick, A.; Gebhard, B.; Härtlein, N.: Modeling of Concurrent Engineering Processes for Integrated Systems Development. Proceedings of the 17th Digital Avionics Systems Conference (17th DASC) Electronics in Motion, IEEE, Bellevue, WA, 31 October - 6 November 1998.

Lullies, V.; Pastowsky, M.; Grandke, S.: Geschäftsprozesse optimieren – ohne Diktat der Technik. Harvard Business Manager, 1998, S. 65-72.

Negele, H.; Fricke, E.; Igenbergs, E. (1997): ZOPH — A Systemic Approach to the Modeling of Product Development Systems. Proceedings of the 7th International Symposium of INCOSE, Los Angeles, 1997

Negele, H.: Systemtechnische Methodik zur ganzheitlichen Modellierung am Beispiel der integrierten Produktentwicklung. Ph.D thesis, Technische Universität München, Herbert Utz Verlag Wissenschaft, 1998

Negele, H.; Fricke, E.; Schrepfer, L.; Härtlein, N.: Modeling of Integrated Product Development Processes. Proceedings of the 9th International Symposium of INCOSE, Brighton, 1999

Biographies

Herbert Negele *is an Assistant Professor at the Institute of Astronautics at the Technical University of Munich. His research focuses on integrated product and process development, and modeling and management of complex systems. He is a founding member of the German Chapter of INCOSE, and was the Vice President in 1997 and 1998.*

Ernst Fricke *is Assistant Professor at the Institute of Astronautics at the Technical University of Munich. His research focuses on managing changes and evolution, process modeling, and the integration of the user into the entire product development cycle. He is a founding member of the German Chapter of INCOSE.*

Nicole Härtlein *is a doctoral student at the Institute of Astronautics at the Technical University of Munich and is working on her Ph.D. thesis in Systems Engineering for BMW. The focus of her work is on process design and documentation in product development. She is currently the Vice President of the German Chapter of INCOSE.*

Modeling Integrated Product, Prospect and Enterprise Development

Jack Ring, jring@kennen.com

Introduction. Product development is too important to be modeled as just process. According to ISO 9000:2000, process is the "system of activities which use resources to transform inputs to outputs." But such a process model does not describe the specifics of resources, inputs, or outputs, and does not specify time, neither elapsed nor anticipatory lead-time. A process model simply describes the logic of the business, not just the chronology, let alone the thermodynamics of the business. Like a functional flow block diagram, the process model describes what is to be done and in what order, but does not describe the result that must be created nor the many implications of enabling and executing the process.

In contrast, product development must be modeled in terms of results and outcomes. The process aspect is necessary but not sufficient because product development must produce results, and those results must be of value to product customers. And, from an internal perspective, beyond the act of product development is the role or purpose of product development. Product development is the seed of wealth building in an enterprise. Accordingly, product development will be successful or not, depending on its cycle time, value added (to its inputs), and return on resources involved. A product development model that does not address at least these factors can be fiction—and fiction that may not be detected until too late. A process-only view can be useful, for example, as an introductory training aid, but is not a sufficient recipe for creating the artifacts and environments that comprise successful products that power successful enterprises. Rather, a project model is required as will be illustrated below.

The "Time to Profit" Perspective. The above assertions may be questioned if one considers only the design and development aspects of product development. But there is more to product development. Time to profit highlights the opportunity of extending the product development envelope in two directions. One axis includes co-developing the product with representative prospective users in order to ensure that the product will be acceptable to the marketplace. The second axis includes creating the new business enterprise that can effectively market, sell and produce the new product and support its new users. This second axis recognizes that new products from old enterprises do not deliver the best value to customers. Rather, every new product deserves a new enterprise. In total, the best results are attained when product, market, and enterprise are simultaneously developed.

If the benefit of a time-to-profit view is acknowledged, then a necessary and sufficient test for its model requires that the model includes all elements and interrelationships that are requisite to success. The necessary test requires that the scope of the model includes the product and the marketplace, as well as the business operations that are required to translate the product into financial gain. The sufficient test requires that the model articulate not just activities, but also how the results of cycle time, value added, and return on resources are to be achieved.

To illustrate this point, this paper introduces a summary time to profit model that integrates product, prospect and enterprise development by showing the major elements, along with informational and temporal relationships. This foundation is then used to highlight the necessary and sufficient semantics required of a results-oriented modeling language.

Model Overview. The Integrated Product, Prospect and Enterprise Development (IPPED) model is summarized in Figure 1. The product development activities are represented in the upper left quadrant, the enterprise development activities are represented in the upper right quadrant and the prospect (market) development activities are represented in the lower half, beneath the belt of standard milestone events that comprise the management framework.

Product Development. Product development starts with a hypothesis regarding market opportunity, solution concept, and market timing (the latter being a scenario of product versions over time, which is documented in the form of a product calendar). These are digested in a subsequent project that culminates in the articulation of the product operations concept (how the product(s) fits in the prospective customers' systems and the value it adds) and product goals (functions, features, performance and price basis). Milestone 1 is reached when the necessary parties concur about the ConOps and goals, and about the key product risks involved in going forward. Risk mitigation action starts with the feasibility prototype, which involves one to three pioneering prospective customers. Milestone 2 is achieved when risk levels are acceptable, or else if risks cannot be mitigated in two prototype cycles, the product is abandoned. If product feasibility is indicated, then accumulated knowledge about the product, the market preferences and the developmental dynamics is fed to the performance prototype project.

A performance prototype, as its name indicates, explores product performance throughout an envelope of operational conditions. The performance prototype is evaluated by three or more "representative customers," whose findings then

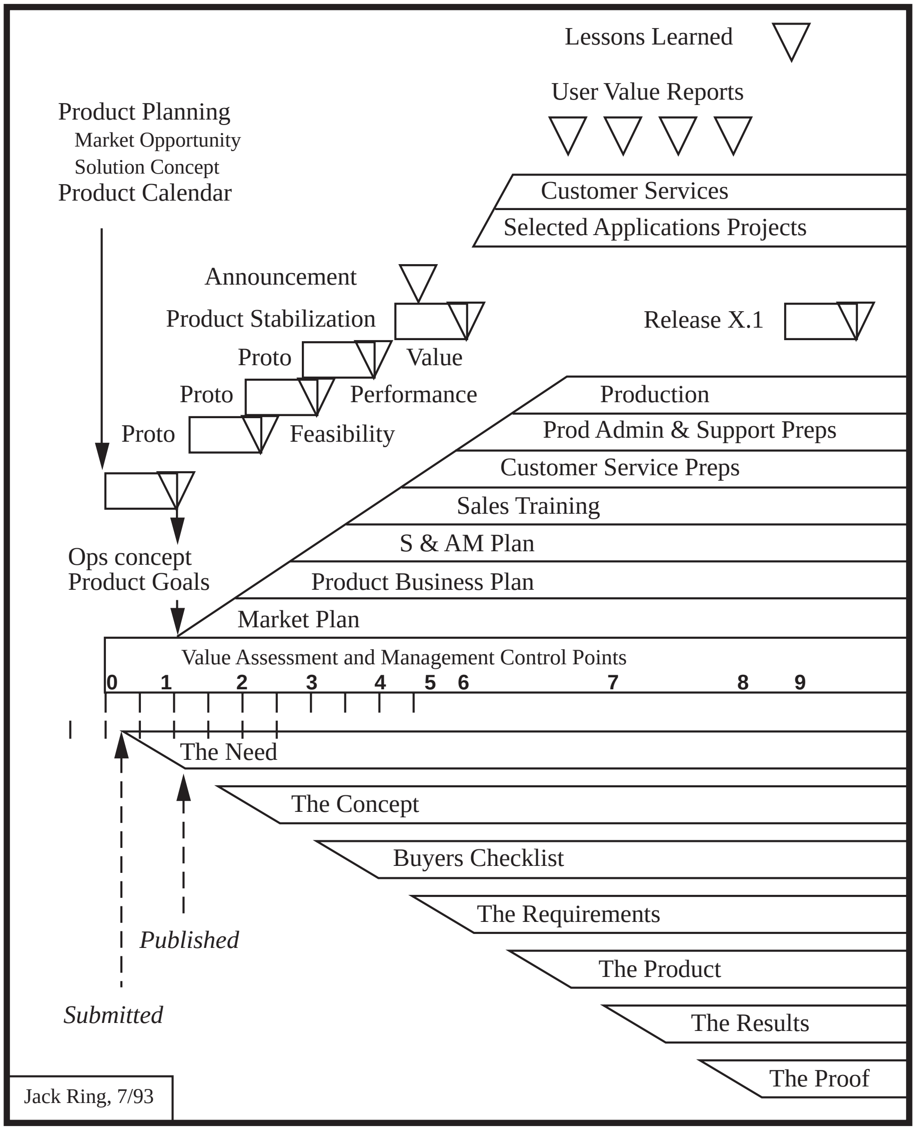


Figure 1. Integrated Product, Prospect and Enterprise Development (IPPED)

inform the value prototype project. Milestone 3 is achieved when three or more evaluations are actually documented. The subsequent value prototype must be installed and operated in a "representative customer" environment because the focus is not so much on the product but shifts to quantifying the benefits that the product brings to the customers' systems. Milestone 4 is achieved when the benefits are quantified. Positive results and loud applause triggers progression to the next project, product stabilization. During the product stabilization project, the business risks are reviewed to verify that their mitigation levels are sufficient to allow product announcement, Milestone 5. A business risk includes product risks as a subsystem but considers market and enterprise risks as well. Completion of the stabilization project permits Milestone 6, controlled release of the product to customers who qualify. A series of customer installations is then supported and user value reports are captured.

Meanwhile, production, which started with Milestone 6, has been refined in structure, process, technologies and behavior until the business risks are acceptable (i.e., anticipated product cost scenario conforms to the business plan and product quality is sufficient and stable). This reduced business risk level enables Milestone 7, general release of the product for sale.

Milestone 8 is reached when sufficient user value reports have made a convincing case for the value of the product to a sufficient number of prospects. Finally, a lessons learned (including a techno-economic model of the product, the last task of the systems engineering activity) enables Milestone 9.

Prospect (Market) Development: Concurrent with the product development projects, a set of seven prospect (market) development projects are conducted to evolve prospects' awareness, appreciation and preference. These are shown in the bottom half of Figure 1. Interestingly, the marketplace's thirst for information is consistent with the pattern of knowledge production that occurs in the product development projects. (Note: A description of the market development and enterprise development projects to the same level as the product development projects is not included in this article. Further details are available from the author.)

Enterprise Development: Now that a path is defined to a good product design and an anxious market, how do we make a profit so we can repeat the cycle? The answer is the stream of seven enterprise development projects that are depicted in the upper right of Figure 1, also positioned temporally with respect to the management framework. These serve to renovate or evolve the existing ways the enterprise interacts with its context in order that it can be prepared to interact properly regarding the new product.

Timing is key: Several of these product, prospect, and enterprise development projects must be orchestrated for an iterative, goal-seeking approach. But the model cannot be a hard-coupled machine. Rather, the model is rubbery in that each project must reflect its real-time context and any one project may affect any other. For example, if product development is conducted too fast the project will not be able to attract "representative customers" fast enough, and may end up designing only what the engineers want to work on. In contrast, if product development projects are too slow, competitors will learn from your initial prospects, then beat you to announcement. Accordingly, the dynamics of the target market – their rate of learning and their level of technology adoption proficiency – informs and paces the product development calendar.

Modeling Integrated Product and Enterprise Development. IPPED is "the application of information to ideas, data and materials to make all more fit for purpose." In this sense, the IPPED is an incremental information generator involving many participants and serving many clients. The model can then be seen as a schema for a dialog among all those involved in making product, process, proficiency and profit. Because dialog is "meaning moving through," this concept accurately characterizes the integrated product and enterprise development model.

Modelers should view IPPED not as a process, but as set of interrelated projects, such as the 24 projects shown in the foregoing illustration, with rubbery interrelations in which any one project establishes the initial conditions for any number of subsequent projects. Each project brings together 1) location vs. time, 2) a process, 3) resources such as products, materials, information and money, and 4) people (notably their competencies, attitudes and interpersonal styles).

Modeling should not be an exercise in fiction writing. A model is not predicting. Modeling consists of making statements of feasibility – that things can happen as described, but are not bound to happen thusly. Modelers keep in mind that results are not produced until process, people, and resources are brought together to interact in real time. Accordingly, each project becomes an emergent system that contributes to the ensemble behavior. If all goes well, the ensemble of 24 projects minimizes enterprise cycle time and maximizes both value (quality) delivered and the return on resources thusly earned. These, of course, are the measures of effectiveness of the system or enterprise.

But what if all doesn't go well? What if the people don't follow the plan? What if the resources are not available as planned and/or the plan is discovered to be infeasible due to risk levels or external events? The answer is to model IPPED as a goal-seeking system (GSS), wherein the model provides not only for noticing such conditions, but also for handling such exceptions. In contrast, most process models are very simplistic and even incomplete, consisting of strings of IF:THEN's with few ELSE's. This is unacceptable, especially when we can expect that successful enterprises will always be operating on the edge of resource availability, thus always encountering constraints. Further, as project experience accrues or new personnel become available, the productivity and innovation coefficients used in the current plan become obsolete and invite a new plan.

If time to profit is to be modeled as a GSS, three aspects must be modeled: a) the nominal operations aspect, b) the orchestration aspect, and c) the alignment aspect. The orchestration project monitors and corrects, or adjusts the nominal operating projects. The alignment project provides for adjusting the plan. Alignment may be triggered by internal findings such as risk mitigation results, or may be triggered by enterprise context changes that call for a change in plan, even if the plan is being executed properly. Both the orchestration and alignment models reveal the precedence rules for handling respective changes, such that the laws of evolutionary optimization are honored. Otherwise, the system of projects can be driven into instability and crash. Similar to configuration management and change control of the product, this is essentially configuration management and change control regarding the overall enterprise and the model of the enterprise.

Modeling IPPED requires the ability to describe each project, and the control and adjustment of each project. Further, the ability to describe the rules for orchestrating the scenario of projects and to estimate the relative effectiveness of each scenario (ensemble of projects) needs to be in terms of the scenario's anticipated measures of effectiveness score. Calculations regarding money, time, and error rates (both too high and too low) are necessary here.

Accordingly, modelers must have the semantics to specify and interrelate the fundamentals — people, locations, artifacts and operations. Interrelation semantics must include elapsed or relative time, and location coordinate systems — not only spatial location but also the cardinal position of any element within its specified coordinate system, such as "M is the thirteenth letter in the English alphabet." Artifacts include stimuli, responses, rewards, and resources incorporated into a response, and the resources that enable the operations (are consumed in or are captured-released by the operation). Similarly, operations are the activities or tasks conducted on artifacts (operands) by operators (such as people, machines, energy, etc.).

It should be noted that the semantics must provide for articulating both the structural and implicit relationships throughout a project and a scenario of projects. For example, the concept of product risk must be representable in the model so risk can be related to the multiple factors (technology competencies, marketplace standards, tasks, need dates, etc.) that comprise the risk. Business risk, of which product risk is only a subsystem, must be similarly modeled. Further, the semantics must provide for typing resources such as money, data, documents and topics. Note that funds, schedules (cash flow), data structures, document trees and lexicons are simply the coordinate systems in which the respective resources are located as a function of time.

Finally, because human beings are the primary appliances in IPPED (the means of making things happen and recognizing that things are not happening) the semantics must provide for characterizing people. Key characteristics are competencies (or proficiencies, including change proficiency), attitudes and interpersonal styles. Also, semantics must provide for characterizing the gradients of these as a function of workload, confusion, and other interpersonal styles existing in the workgroup.

No known modeling language embodies this fundamental set of semantics. It can be argued that many existing modeling languages can be extended to this end but in the author's experience when IDEF, UML and other languages in the functional decomposition family are thusly extended, the modeling tool becomes more complex than the system being modeled. Relational modeling languages such as RDD-100 and CORE similarly tend to complexity too rapidly. Promising modeling language candidates are based on formal ontologies that define multiple relationship types and allow context dependent relationship expressions. One example is the holon-based approach [1, 2]. Others that may evolve to sufficiency include ZOPH [3], IDEON™ [4] and eManagement.net™ [5]. The OOCL [6] extensions to UML, if adopted, may improve the UML situation considerably.

Conclusion: Integrated product, prospect and enterprise development, (and especially its systems engineering content) becomes clearly necessary or beneficial when:

Mission: The enterprise mission is to provide products, information and services that add value to a customer's system and earn rewards (e.g. revenue) that establish a gross profit stream.

Situation: Enterprise success is not guaranteed by just the creation of an excellent new product but involves gaining customers, as well. Product perpetrators must help prospective customers become aware of the new product, appreciate the value-enhancing opportunity the new product offers them, and arrive at a preference for the product as compared to competitors' alternatives or to doing nothing.

Scope: Enterprise not only must create the product but also must understand the meaning of the product and be proficient in marketing, sales and user support. They must ensure that production, administrative and other business processes are responsive to both the properties of the new product and the business responsiveness demanded by customers.

Exploratory Approach: Because new and unique products do not find ready markets, both product and business processes must be prototyped and evaluated in anticipated operational settings with representative customers until both producer and representative customers can quantify both solution and value added. This means that product, prospect, and enterprise development are best done by an iterative method with a goal-seeking attitude. Further, the activity cannot be accomplished in a secret laboratory. The iterations, or at least a large percentage of them, must be conducted in situ.

Context Stability: When the mean time to change of external pressures on the enterprise, such as changes in markets, economics, regulations and technologies, is shorter than the time to profit cycle.

Reflection: A model communicates a vision. If all you model is process then all you will get is process. A cynical view of process says that by complying with ISO9000 you can go broke in an orderly, documented way. Conversely, if you model time to profit, then you have a fair chance of gaining profit — quickly.

References:

Business Threads™, Koeslon, Inc., http://www.koeslon.com

Attricom™, Hologix, Inc., http://www.hologix.com

ZOPH – A Systemic Approach to the Modeling of Product Development Systems, Negele, Herbert, Ernst Fricke and Eduard Igenbergs, Proceedings of the 7th Annual Symposium of INCOSE, 1997

IDEON™, Intelligent Systems Technologies, Inc., http://www.intelligentsystems.com

Enterprise Strategist™, Enterprise Software Systems Inc., http://www.enterprisestrategist.com

OOCL, The Knowledge Management Consortium, Inc., http://www.km.org

Biography:

Jack Ring, *concurrent co-founder of Innovation Management, Kennen, Inc. and Ethos Hospitality, Inc., consults internationally by applying the principles and practices of systems thinking and knowledge management to market intelligence, enterprise strategy, business results engineering, product design, and staff enthusiasm.*

Process Modeling with Design Structure Matrices (DSMs)

Tyson R. Browning, tyson.browning@lmco.com

Everyone wants better products faster and cheaper. Providing such products requires lean, efficient, capable, robust, and value-adding product development processes. In the last decade, some have emphasized lean manufacturing as part of the solution. But before production, product design activities determine most of a product's life cycle value. To highly impact value, we must better understand product design activities and how they work together.

Product design processes are unlike typical business and production processes in several ways. Product design processes are described by terms like "iterative" and "creative." They entail a number of disciplines proposing, analyzing, and negotiating to find mutually acceptable solutions. At a detailed level, things may not happen the same way twice. How can we better understand such processes and their constituent activities? How can we model something as complex and seemingly unstable as a system development process in enough detail to really analyze and improve it?

First, we must discern what to model or describe. Product development is a process of information collection, interpretation, creation, transformation, and transfer. Product development activities require and produce information. Performance risk is reduced as new information is created. To achieve their maximum value, activities must execute based on the right information, which they must have at the right time. Thus, the value of activities is to an extent a function of the value of the information they create. Lack of value may stem less from doing the wrong activities, than from doing the right activities with the wrong information (and then having to redo them). Fortunately, we can model the product development process by describing the information it produces, much of which flows among activities in the process. More generally, we can model the dependencies among activities, many of which involve information.

Modeling dependencies is preferable to modeling workflow for several reasons. When we model workflow, we "hard wire" a particular ordering of activities. We mandate what will happen sequentially and what will occur concurrently. Usually, we describe the way things have always been done (which probably needs to be improved) or the way we would like things to be done (which is unverified and may be infeasible). We need a process description that affords more flexibility for both planning and replanning. An activity dependency model describes what each activity needs to do its job and where that input comes from. Dependence is a firmer constraint than an arbitrary workflow decision. We can analyze a dependency model to prescribe an efficient and feasible workflow. We can plan and replan a workflow based on what data (and other resources) are needed and available.

In the early stages of product development, most of the dependencies among activities signify reliance on information. Information dependencies are relatively more stable than activities. In creative processes, we may not do a given activity exactly the same way twice. However, an activity is likely to have to produce a specific type of information, such as "Power Requirements," every time. It may be more useful to model the information products created and required throughout the process than to dwell on the less stable collection of activities providing and consuming them. Actually, we need to model both. The problem is that most business process modeling focuses on activities and workflow and gives inadequate attention to the full range of dependencies among activities.

This article describes a method for process modeling based on the dependencies among process elements. The method, called the *design structure matrix* (DSM[1]) and, codified by Donald Steward in 1981, stems from matrix algebra and precedence diagram work in the 1960s. The DSM is similar to the N-square diagram—a familiar systems engineering tool used to represent system elements and their interfaces—with the addition of a time basis.

Describing the DSM

As shown in Figure 1, a DSM is a square matrix with corresponding rows and columns. The diagonal cells represent the activities, which are listed from upper left to lower right in a roughly temporal order. Off-diagonal cells indicate the dependency of one activity on another. Dependencies are often needs for information. Reading down a column shows information sources; reading across a row shows information sinks.[2] For example, Activity 1 provides information to Activities 2, 4, 5, and 6. Activity 2 depends on information from Activities 1 and 6 and provides information to Activities 3 and 4.

Figure 1 shows how the DSM displays dependent, independent, and interdependent activity relationships. Since Activity 2 depends on information from Activity 1, these two activities will probably be executed sequentially in the workflow. Activities 3 and 4 do not depend on each other for information, so they may safely proceed in parallel (barring other resource constraints). Activities 5 and 6 both depend on each other's outputs. These activities are said to be interdependent or coupled and are discussed below.

Of particular interest are the cases where marks appear in the lower triangular region of the DSM. Such marks indicate the dependence of

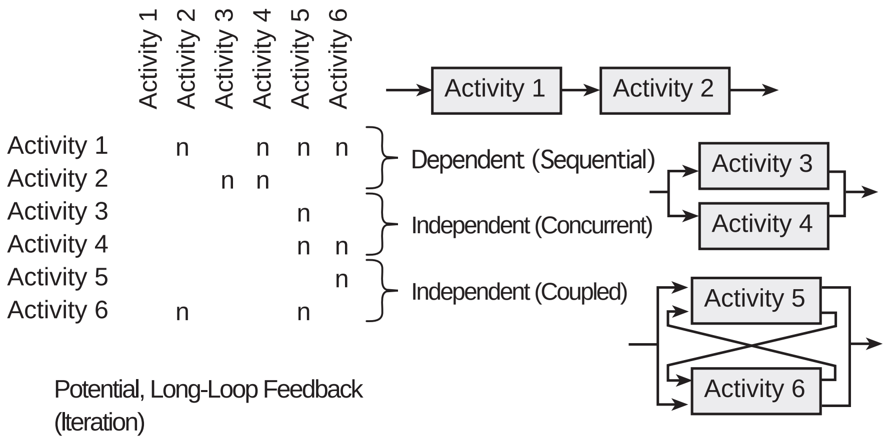


Figure 1: Example DSM

an upstream activity on information created downstream. If project planners decide to execute the activities in the given order, Activity 2 will have to make an assumption about the information it needs from Activity 6. After Activity 6 finishes, Activity 2 may have rework if the assumption was incorrect. The DSM conveniently highlights iteration and rework, especially when it stems from activities working with potentially flawed information.

When we see a mark in the lower left corner of the DSM, we know that there is a chance of having to return to the beginning of the process, which could have a catastrophic impact on cost and schedule. The marks in the lower left corner of the DSM may represent key drivers of cost and schedule risk. Rearranging the activity sequence (by rearranging the rows and columns in the DSM) can bring some subdiagonal marks above or closer to the diagonal, thereby reducing their impact. Simple algorithms automate this exercise. Adding quantitative information to the DSM and using simulation can quantify the impacts of process configuration changes on cost and schedule risk.

Sometimes a subdiagonal mark cannot be brought above the diagonal without pushing another mark below the diagonal. This is a case of interdependent activities, such as Activities 5 and 6. Each activity depends on the other. They must work together to resolve a “chicken and egg” problem. Typically, coupled activities work concurrently, exchanging preliminary information frequently. If a subset of coupled activities must begin before the rest, the more robust (less volatile and/or sensitive) information items should be the ones appearing below the diagonal in the DSM. If coupled activities are functionally based, an opportunity may exist to fold the activities into a single activity assigned to a cross-functional team.

Integration, test, and design review activities typically have marks in their rows to the left of the diagonal. These activities create information (including results of decisions) that may cause changes to (and rework for) previously executed activities. Unfortunately, most process planners “plan to succeed” and their process models fail to account for these possibilities. Fortunately, the DSM provides an easy way to document potential “process failure modes” and their effects on other activities. The simple marks in the DSM can be replaced by numbers indicating the relative probability of information change, iteration, etc. This enables an analysis of process failure modes and their effects on cost, schedule, and risk. Process improvement investments can then target mitigation of the biggest risk drivers.

As a real-life example, Figure 2 (following page) displays a DSM of the Conceptual and Preliminary Design phases for an *uninhabited combat aerial vehicle* (UCAV).[3] The first dozen activities comprise the Conceptual Design phase. In this phase, design requirements and objectives (DR&O) are prepared, a configuration concept is proposed, it is analyzed by a variety of discipline perspectives, and then these results are assessed. The assessment may reveal a need to alter the DR&O, to create a new configuration concept, and/or to alter the current configuration concept. This cycle repeats until the design space is sufficiently understood and/or time and money are exhausted. The design process then moves into the Preliminary Design phase, where the configuration is developed and analyzed in more detail and the objective is to prepare a proposal to acquire funding for additional phases. Figure 2, shows the process “as is,” without any attempt to resequence the process to eliminate feedback. This basic model served as the basis for additional process analysis, evaluation, discussion, and improvement.

The DSM simply provides a view of a process model that exists as a collection of two kinds of objects: process elements (e.g., activities) and data packages. Process elements have characteristics such as inputs, input sources, outputs, output destinations, duration, required resources, etc. Data packages have creators, consumers, and other attributes. The DSM shows how these objects link together to form a process of information creation and consumption such as product development.

The process element objects will often be formed in a decentralized fashion. For example, a survey can be sent to a person with expertise regarding a particular activity. The survey may ask the person what things are needed to perform the activity, where those things come from, what things are produced, and where those things go. A model integrator then assimilates multiple process element survey responses. The model integrator will spend most of his or her time resolving the number and names of the data packages linking the process elements. Despite this effort, a somewhat

Activities		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Prepare UCAV Conceptual DR&O	1		n	n	n	n	n	n	n	n	n	n	n															
Create Configuration Concepts	2	n		n	n	n					n	n	n															
Prepare 3-View Drawing & Geometry Data	3				n	n	n		n																			
Perform Aerodynamics Analysis & Evaluation	4						n	n		n	n	n		n		n												
Perform Propulsion Analyses and Evaluation	5				n		n	n		n	n	n		n		n												
Perform S&C Characteristics Analyses & Eval.	6										n	n		n		n												
Perform Mechanical & Electrical Analyses & Eval.	7					n			n		n	n		n		n												
Perform Weights Analyses & Evaluation	8			n	n		n			n	n	n	n	n		n				n								
Perform Performance Analyses & Evaluation	9										n	n		n		n												
Perform Multidisciplinary Analyses & Evaluation	10											n																
Make Concept Assessment and Variant Decisions	11	n	n										n															
Prepare & Distribute Choice Config. Data Set	12													n	n													
Prepare UCAV Preliminary DR&O	13														n		n	n	n		n	n			n	n		n
Create UCAV Preliminary Design Configuration	14															n												n
Prepare & Dist. Surf. Models & Int. Arngmnt. Drwings.	15																n		n		n							n
Create Initial Structural Geometry	16																	n								n		n
Prepare Structural Geometry & Notes for FEM	17																n			n		n	n	n	n			n
Perform Aerodynamic Analyses & Evaluation	18															n					n		n	n				n
Perform Weights & Inertias Analyses & Evaluation	19																n				n		n					n
Perform S&C Analyses & Evaluation	20														n													n
Develop Structural Design Conditions	21																						n	n	n			n
Develop Bal. Freebody Diagrams & Ext. App. Loads	22																							n	n			n
Establish Internal Load Distributions	23																						n		n			n
Evaluate Structural Strength, Stiffness & Life	24																n			n						n		n
Evaluate Plan Manufacturing & Tooling	25																n										n	n
Create Resource Tables & Evaluate Cost	26																n											n
Prepare UCAV Proposal	27																											

Major Cycles of Conceptual Design

No return to Conceptual Design from Preliminary Design

Figure 2: DSM of Conceptual and Preliminary Design Phases for a UCAV

decentralized approach allows the people doing the work to contribute to the model, yielding a more accurate process description that users will accept.

Conclusion

The DSM provides a concise, visual format for representing processes. A process flowchart consuming an entire conference room wall can be reduced to a single page DSM. After a quick orientation, everyone can see how his or her activity affects a large process. People can see where information comes from and where it goes. They can see why delaying the activities they depend on forces them to make assumptions, which may trigger rework later. It becomes apparent that certain information changes tend to cause rework. Such situation visibility and awareness leads to improved process design and coordination. The DSM also provides a process knowledge base from which the foundations of process plans and risk assessments can be drawn. Moreover, the DSM is amenable to some simple yet powerful analyses.

DSMs have been developed for planning and managing projects in the building construction, photographic, semiconductor, automotive, aerospace, telecom, and electronics industries. More detailed and quantitative models based on the DSM have been developed by several researchers. Methods are under development for structuring nested, hierarchical DSMs.

Perhaps the greatest barrier to DSM usage is the amount of information required to characterize the structure of a design process. DSMs representing complex system development processes call for integrating the expertise of a number of people. Building a DSM also forces some people and groups to think in terms they may not be accustomed to. But this is good, and it should be made to happen anyway. A great amount of benefit is often realized simply by participating in the DSM construction process.

For more information on DSMs, please contact the author.

Footnotes:

1 a.k.a. dependency structure matrix

2 Some DSMs use the opposite convention—rows for sources and columns for sinks—resulting in feedback appearing above the diagonal. The two conventions convey equivalent information.

3 The UCAV example comes from The Boeing Company and is fully documented in (Browning 1998).

References

Browning, Tyson Rodgers (1998) Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development, Unpublished Ph.D. Thesis (TMP), Massachusetts Institute of Technology, Cambridge, MA.

Eppinger, Steven D., et al. (1994) "A Model-Based Method for Organizing Tasks in Product Development" Research in Engineering Design, 6: 1-13.

Steward, Donald V. (1981) Systems Analysis and Management: Structure, Strategy, and Design, New York: PBI.

Biography

Tyson Browning conducts applied research and provides internal consulting on engineering process development for Lockheed Martin Tactical Aircraft Systems in Fort Worth, Texas, USA. He previously worked with the Product Development Focus Team of the Lean Aerospace Initiative at MIT.

A Model-Based SEMP Approach

Rick Steiner, fsteiner@west.raytheon.com and Doug Stemm, djstemm@west.raytheon.com

Introduction. This article describes the development and deployment of a Systems Engineering Management Plan (SEMP) model on an ongoing project. At the start, this project required a considerable amount of innovation, both in product technology and in distributed collaborative techniques. A traditional paper-based process description quickly proved inadequate to the task. The project also required development and deployment of a new collaborative infrastructure, which had to be compatible with the project's systems engineering process.

Process Models. Methods for formally modeling processes have matured over the years, particularly as applied to improving production efficiency and workflow. Unfortunately, the use of process models as basis for project planning and management of complex system development has not been so successful. While the reasons for this lack of success vary, the potential benefits of an integrated process model are great enough to justify further attempts. The benefits of such an integrated model include:

- Clear, hierarchical definition of task structure, with control and data interdependencies
- Definition of work product structure, with clear traceability to tasks and relationships to each other
- Definition of team structure, and unambiguous allocation of tasks to teams
- Definition of tools and procedures, with clear allocation to tasks

Modeling the SEMP. A SEMP should include: 1) unifying or common top level processes, 2) description of deliverables, 3) description of organization structure, 4) description of top level tasks, 5) allocation of tasks to teams, and 6) description of reviews and review authority. The intent of the SEMP model was to integrate all six of these elements.

RDD-100 was selected as the modeling tool because of its ability to render multiple graphical notations, and its robust linking and report writing capability. Tasks were represented as "discrete functions" in RDD, and aggregated into "time functions" to provide a task hierarchy. Work products were represented as "discrete items," and aggregated into "time items" where appropriate. These work products were also classified as deliverables, guidelines/constraints, design information, or requirements. The top level IDEF0 diagram for this model is shown in Figure 1 below.

Note that IDEF0 is one of several functional notations that RDD can render, on the fly, from the process model. Others include behavior diagrams, functional flow block diagrams, and data flow diagrams. IDEF0 is used to convey the content of the process model because it is well documented and generally accepted in the process engineering community, while the other notations tend to be restricted more to the systems engineering community.

Note that the model is built hierarchically, with each tier of decomposition organized with a manageable number of elements. The small black square in the upper left corner of a function indicates that it contains

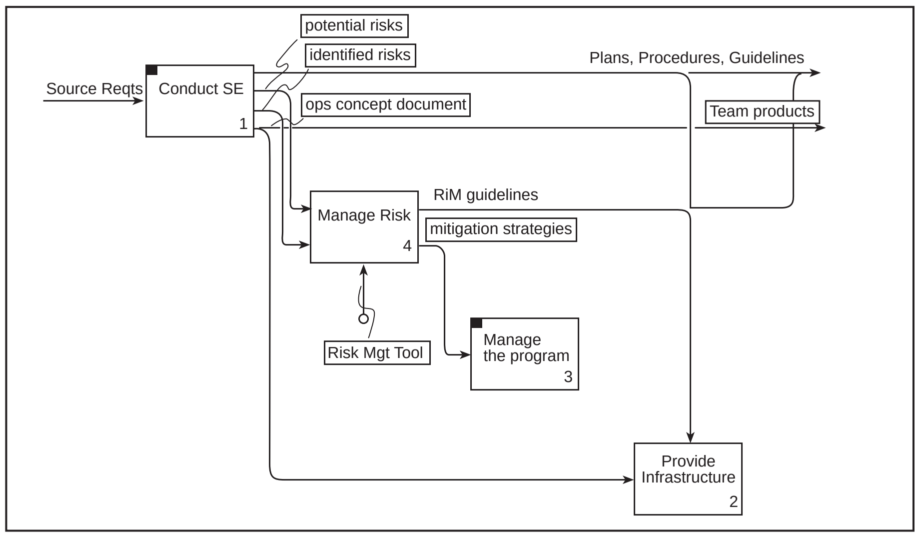


Figure 1: Top Level SEMP diagram (IDEF0)

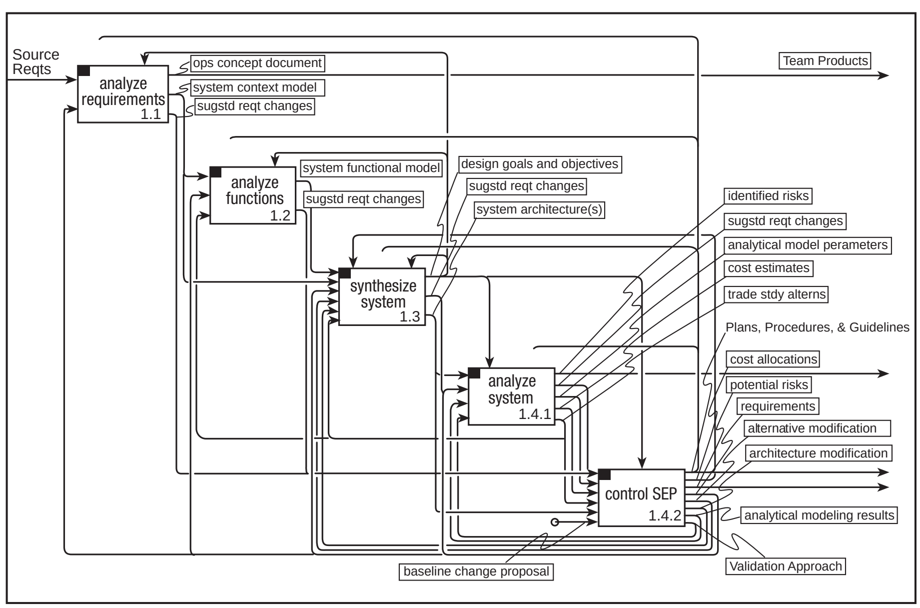


Figure 2: Conduct Systems Engineering

further levels of decomposition. The next level of decomposition for "Conduct Systems Engineering" is shown in Figure 2.

Tasks have been broken down only far enough to be uniquely allocated to a single team, or to identify how they contributed to a key deliverable. It is inappropriate to attempt to model individual human behavior, or the rationale behind key technical decisions. It is enough to identify that key technical decisions to be made, the specific teams that will make them, and that the resultant data will be captured in the appropriate tool or database. While dynamic execution of this model has not yet been attempted, it will at most be used only to identify work product timeline conflicts, and will NOT be used to generate realistic program plans. It is possible that the task elements and dependencies in this model may be used to drive a program planning tool, such as Microsoft Project, and thus generate an Integrated Master Schedule (IMS). Program planning and status is not the main objective of the SEMP model, however.

Infrastructure elements necessary to the project (tools, networks, legacy databases) have been captured as "components" in RDD. Allocation of the tasks to individual tools resulted in the tools appearing on the IDEF diagrams (e.g. the Risk Management Tool in Figure 1). Interfaces between tools and databases have been depicted as "item links" in RDD, and are visible in the RDD "component diagram." Because the tools and databases were mapped to tasks, and inputs and outputs of tasks were already defined, the definition of data passing between tools and databases became straightforward. Data element definition within work products (task inputs and outputs) was performed using "abstract object types" in RDD.

The project team structure has been captured hierarchically as "real world objects" in RDD, and each team has been allocated their appropriate tasks. Once again, task inputs and outputs are then readily identified, this time by team. This begins to demonstrate the usefulness of an integrated process model.

To disseminate the information developed in the model, it was decided to create a SEMP web page accessible to all project participants and customers.

IDEF0 diagrams from RDD have been extracted as postscript files, and converted to pdf files using Adobe Acrobat Exchange. Infrastructure element relationships in component diagrams have likewise been made available as pdf files. Linkages between diagrams have been manually established, resulting in a navigation capability similar to that within RDD. (It would have been very helpful if RDD could export the pdf files directly, along with the linkages between them.)

Hierarchy diagrams for team structure, tool/database structure, and work product structure have also been captured in pdf format. Task allocation to teams, and the resulting deliverable responsibilities, have been extracted using the "Multi Element View" capability in RDD, and converted to pdf format as well.

Conclusion. The SEMP web page has helped focus the systems engineering process across the widely distributed project team. It is anticipated that the SEMP model will continue to grow in complexity, but at the same time continue to tightly link the individual program elements. It is hoped that as the program progresses, individual teams will rely on the SEMP model as a vehicle for process understanding and innovation.

Biographies:

Rick Steiner *has worked for Raytheon Systems Company (formerly Hughes) for the past 15 years, applying systems engineering processes and tools to naval and maritime electronics systems development. He has been a member of INCOSE since 1993, and has participated in various working groups of the Modeling and Tools Technical Committee.*

Douglas J. Stemm *is System Engineering Process Owner and Manager of Requirements Management / Generation Lead for Naval and Maritime Systems (NAMS), a business unit of Raytheon Systems Company's Defense Systems Segment. He joined Hughes Aircraft Company in 1974 (Hughes sold to Raytheon in 1997) and has 26 years of experience as a software engineer, system engineer, and associate program manager in various air defense systems, sonar systems, and command and control systems.*

Note: RDD-100 is a registered trademark of Ascent Logic Corporation. Adobe Acrobat is a registered trademark of Adobe Systems Incorporated.

The Application of System Dynamics to Concurrent Engineering

David N. Ford, David.Ford@ifi.uib.no

Introduction. The design manager appeared simultaneously exhausted, depressed and frustrated. The project manager had been forced to postpone the deadline of the development project again. The delay wiped out the benefits of the concurrent engineering approach that was intended to bring their product to the market before their competitors. Even though design changes had precipitated the slip, the design manager knew he had not been the real cause.

Not long after design work started his estimated dates to complete and release the design packages had exceeded the deadlines he had negotiated with the managers of the project's other phases. He was tempted to ask for more time, but he remembered the project manager's exhortation for proactive management to keep the project on its aggressive schedule. So he had initiated overtime and started recruiting more designers.

But during the time needed to find and train the new designers, the existing staff spent more time training and less time designing and got "burned out" trying to maintain the scheduled rate of progress. This created a big backlog of unchecked work and unresolved design changes, and extended his expected completion dates even further. To meet the "deliver or else" deadlines to release designs to the prototyping and production planning phases, he had reluctantly released designs without complete testing and hoped for the best. But his reprieve from schedule pressure was only been temporary. As the prototype and production engineers tried to use the designs they discovered errors. His "Changes to be Coordinated and Resolved" file had grown much faster than his staff could address it. The estimated completion dates stretched out even further into the future. When the project deadline neared and the project manager intensified his questioning at the weekly managers meeting, the prototype and production planning managers mentioned waiting for these changes as the cause of the delays in their own phases. It was then that the project manager had been forced to slip the schedule, with a menacing glare at the confused design manager.

This hypothetical but typical dilemma illustrates some of the challenges to successfully implementing concurrent engineering in development projects. The diversity of actions and agents, and the relationships that link them is an important feature of concurrent engineering projects. The story above includes development operations (e.g. design, quality assurance and coordination), resources (design staff), management (deadlines and policies), and the behavior of the participants (shifting blame). The diversity and tight coupling of project components make understanding and improving concurrent engineering impossible by focusing solely on development operations, or resources, or management, or the behavior of participants. Only by modeling and analyzing development operations, resources, and management, and the behavior of participants and how they interact, can the systems nature of concurrent engineering be understood and improved. Designers and managers of concurrent engineering projects need to understand how different components influence each other in order to design changes that will improve performance. The effective modeling and analysis of concurrent engineering requires an approach that can describe a variety of components and relationships.

One of the challenges in modeling concurrent engineering is that the important causal paths that link design and management to performance pass through several of the diverse subsystems of the project. For example, an explanation of why the project described above was delayed snakes its way through schedule targets, estimates and pressure, design operations, design management policies about overtime and staffing, design quality and release policies, concurrence between phases, prototype and production quality assurance operations, interphase coordination and change resolution and project schedule flexibility. Effectively diagnosing the drivers of concurrent engineering requires explicitly modeling the locations and characteristics of the causal paths that describe how project components affect each other.

Structural feedback is a dominant characteristic of the causal paths in concurrent engineering projects. Structural feedback exists when the impacts of a change in a component travel through one or more causal paths and return to influence the component. In the story above, the impacts of the schedule pressure caused the design manager to recruit designers who influenced the effective size of the design staff, and thereby the design backlog and amount of schedule pressure. Structural feedback is inherent in concurrent engineering due to its many strong dependencies. This makes concurrent engineering projects dynamically complex, evolving over time as project participants, respond to conditions, and those conditions respond to attempts to control the project. Policies that improve conditions initially (e.g., releasing unchecked designs) can degrade performance as delayed effects play themselves out. Systems engineers need a dynamic perspective to design and analyze processes, and policies that generate different behaviors over time.

The System Dynamics Approach to Concurrent Engineering. System dynamics (Forrester 1961) applies a unique perspective and modeling methodology to concurrent engineering. Three central system dynamics concepts for application to concurrent engineering are structural feedback, delays, and participant behavior. System dynamics identifies structural feedback at the operational level that can generate observed or desired patterns of behavior over time. This approach contrasts with an assumption that external forces are the primary causes of behavior. The focus on internal causes of behavior helps identify high leverage points that can be used to improve performance, and unintended side effects that, in turn, can defeat management efforts. An example of an unintended side effect due to the design manager recruiting more designers is the increase in training and resulting decrease in effective design staff size.

System dynamics also focuses on the delays that constrain progress and distort information. The development operations and management of concurrent engineering projects generate many delays. With structural feedback, these delays cause the long-term impacts of some policies to be very different from their short-term impacts. For example, the short term effect of releasing inadequately checked designs was to reduce schedule pressure, but the delayed effect was to increase design backlog and increase schedule pressure. A feedback perspective and explicit modeling of delays is particularly valuable in modeling management policies and the behavior of participants. Management policies describe the use of system conditions and decision rules to generate actions designed to control behavior. Participant behavior, such as the fatigue experienced by the design staff, strongly influences the effectiveness of policies. The mental models and limits of cognition of both policies and participant behavior perturbs the process and therefore must be described with nonlinear relationships.

Applying system dynamics to concurrent engineering requires effective communication with managers and developers who provide model information, develop insights, and implement system changes based on modeling and analysis. In many cases, these critical participants in the modeling and analysis are unable to understand or use mathematical models. To address this constraint, system dynamics has developed modeling tools and methods that are effective with non-technical participants, as well as those experienced and comfortable with formal models. A typical system dynamics modeling project uses causal loop diagramming (Richardson and Pugh, 1981) to describe the system operations related to the problem and current management. Causal loop diagrams describe the feedback structure of a system and locate major delays.

Figure 1 shows a causal loop diagram that maps part of the feedback structure of the story at the beginning of this article. Balancing loop B1 describes how the design manager's initial response to schedule pressure of recruiting designers was intended to reduce the backlog and schedule pressure. But the unintended side effect of increased training requirements described by reinforcing loop R1 defeated this policy due to the use of experienced designers for training and the delay in getting new designers productive. This reduced the effective size of the design staff, thereby increasing the backlog and schedule pressure. Balancing feedback loop B2 describes the design manager's second policy to reduce schedule pressure, releasing unchecked designs, and its initial success by reducing the design backlog. Reinforcing loop R2 describes how the delayed downstream discovery of change requirements increased the design backlog, increasing the schedule pressure instead of decreasing it, and thereby precipitating the slipping of the project deadline.

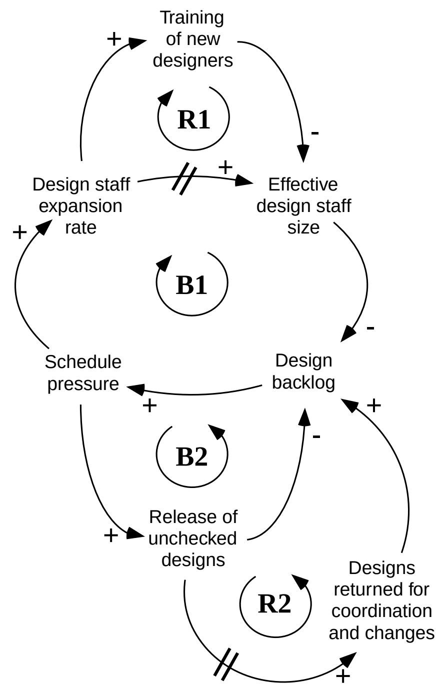


Figure 1: A Causal Loop Diagram of Managing Schedule Pressure

Legend

—-> - a causal relationship: change in variable at arrow's tail causes change in variable at arrow's head

+ - variable at arrow's head moves in the Same direction as the variable at the arrow's tail

- - variable at arrow's head moves in the Opposite direction as the variable at the arrow's tail

// - delay in response of variable at arrow's head to change in variable at arrow's tail

R - Reinforcing feedback loop: in isolation generates exponential growth or decay

B - Balancing feedback loop: in isolation generates controlling or goal-seeking behavior

Feedback Loops

B1 - Staff size adjusts with a delay in response to schedule pressure

R1 - Design training increases backlog and schedule pressure, increasing staff and training

B2 - Release of unchecked designs responds to schedule pressure

R2 - Delayed return of designs increases schedule pressure, increasing release of unchecked designs

Identifying feedback loops, such as the ones described above, is necessary, but not sufficient, to improve concurrent engineering project design and management. System dynamics formalizes feedback models into mathematical descriptions of the flows and accumulations of work, people, and information in concurrent engineering development projects. These sets of differential equations allow computer simulation of the behavior generated by the system as described, and hypotheses about what structures drive behavior to be tested.

The variation in the types of system components and their relationships in system dynamics models requires a variety of tests to develop confidence in the model's ability to simulate realistic behavior for the same reasons that the behavior is generated in real projects. Examples of these tests include the direct comparison of the model structure with the structure of the project, tests of model behavior for realistic responses over a wide range of conditions, and comparisons of simulated and actual project behavior. The model is ready for use as an analysis and design tool when it simulates actual project behavior patterns using the same causal paths as actual projects.

Applying System Dynamics to Concurrent Engineering: An Example. One example of applying system dynamics to concurrent engineering is Ford (the author) and Sterman's (1998b) investigation of the relationship between concurrence and project schedule performance. Our model describes how the initial completion of work, quality assurance, changes and coordination move work through six conditions in individual development phases (Figure 2). Individual phases are linked with concurrence relationships, coordination and performance targets. Management policies, such as the use of unchecked work as the basis for additional development, work release package sizes, resource allocation policies and the flexibility of schedules, budgets and quality goals, are explicitly modeled so that their descriptions can be discussed with and verified by practitioners, and so that the impacts of policies can be tested.

We calibrated our model to a medium sized semiconductor development project. Data collection included running workshops to elicit and articulate the expert but tacit knowledge about concurrence relationships within and between development phases held by the development engineers and managers. The developers found these workshops useful because they were able to share and compare mental models in a forum which facilitated learning by investigating their underlying assumptions (Ford and Sterman, 1998a).

By explicitly modeling the concurrence relationships within and between development phases, and the constraints on progress imposed by development operations and the resources applied to them, we could analyze the conditions under which each part of the project drove or constrained progress. We used the model to study how unintended side effects of increasing concurrence can defeat efforts to shorten cycle time with concurrent engineering (Ford and Sterman, 1999). We did this by focusing on the causes of the "90% syndrome," a common form of schedule failure in concurrent development. We found that increasing concurrence and the common propensity of developers and managers to conceal required changes from other development team members aggravates the syndrome and degrades schedule performance through feedback loops that control iteration between project phases. Based on this analysis, we suggested iteration management strategies that may improve concurrent engineering implementation.

System dynamics models can compliment other systems engineering approaches to concurrent engineering. Consider the potential synergy between Ford and Sterman's system dynamics model and Melsa and Smith's (1998) approach to defect control for concurrent software development. Melsa and Smith describe a set of policies for controlling the number and release of defects to subsequent development phases and customers. For example, one policy uses system conditions (e.g., the number of unresolved defects) to allocate resources to development operations. They initially test their approach by applying it to a single development project. Despite their success many questions remain

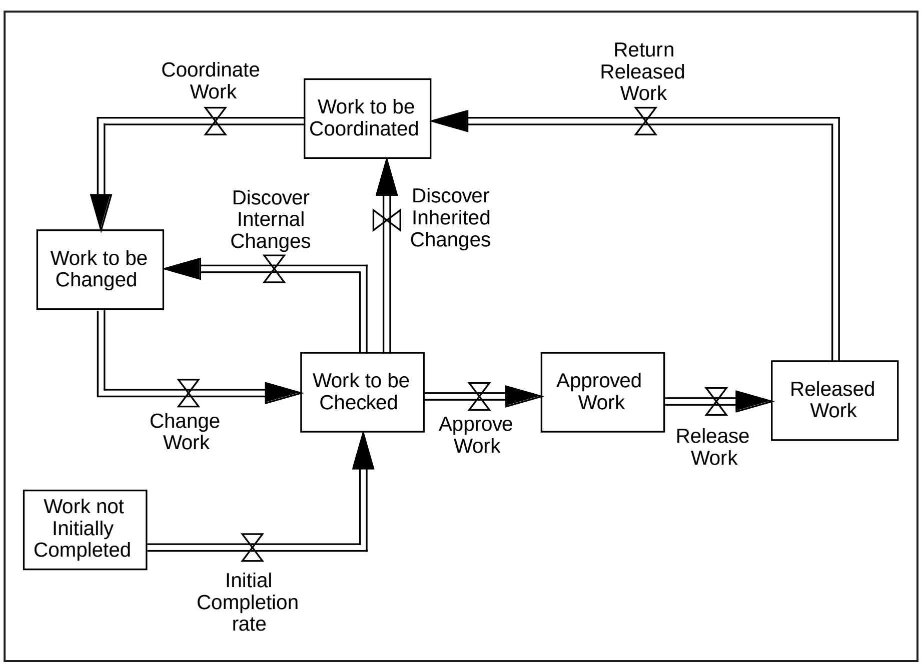


Figure 2: Flows and Accumulations of Work in a Single Development Phase

before their approach can be fully understood and applied to other projects, which may differ in important ways from their case study. "How sensitive is the project's performance to specific project characteristics such as size and complexity? Which policies and combinations of policies increase performance most under what conditions?" More importantly for developing widely applicable insights from their work, "How can we model and test the recommended or alternative policies in the iterative environment that characterizes concurrent engineering and under a variety of conditions?"

A system dynamics approach can be used to develop just such a tool. For example, Ford and Sterman's model includes all the important aspects of concurrent engineering identified by Melsa and Smith. In addition, the majority of the variables proposed by Melsa and Smith as useful for practicing managers are included in the formal mathematical model. The simulation model facilitates the discovery of previously unknown defects, which can be explicitly modeled with the known defects. The model also allows experimentation and exploration of concurrent engineering project structures and policies that could not be done with actual projects due to costs, feasibility and other constraints. System dynamics based project simulation models can reflect the features and issues addressed by many other systems engineering methods, and provide valuable opportunities for analysis and design not available by other means.

Conclusions. System dynamics is particularly effective for modeling and analyzing how the interactions of structures and policies impact project performance in concurrent engineering projects. This is partially due to its focus on behavior generated by structural feedback, multiple time perspectives and the ability to equally model development operations, resources, management, participant behavior and their interactions at an operational level. The method's flexibility allows the rigorous modeling of interactions among the variety of subsystems which form concurrent engineering projects. This makes system dynamics particularly well suited for systems engineering, which addresses the design and management of the interfaces where those interactions occur.

The flexibility of system dynamics provides both great strength as a modeling methodology, and special challenges in developing confidence in the model's ability to reflect projects accurately, and rigorously analyze and describe how project structures drive behavior. System dynamics has been applied to investigate concurrent engineering projects in shipbuilding, consumer and industrial electronics, semiconductor development and, most recently, the construction industry. The construction industry in particular, where concurrent engineering has been practiced for over two decades in fast-track projects, provides unique opportunities to discover the fundamental drivers of behavior in concurrent engineering projects, and how to manipulate them to improve performance.

System dynamics provides a perspective and modeling method that can be used to investigate the interactions of concurrent engineering development operations, resources, management, and participant behavior at the operational level. Its ability to elucidate and explain how the complex structures of concurrent engineering projects drive performance makes it a valuable systems engineering tool.

Biography:

David Ford *is an associate professor in the system dynamics program at the Department of Information Science, University of Bergen, Norway. His research investigates the design and management of development projects.*

References

Ford, D. and Sterman, J., Expert Knowledge Elicitation for Improving Formal and Mental Models. System Dynamics Review. 14:4:309-340. Winter, 1998 (a).

Ford, D. and Sterman, J., Overcoming the 90% Syndrome: Iteration Management in Concurrent Development Projects. Working paper. Sloan School of Management. Massachusetts Institute of Technology. Cambridge, MA. 1999.

Ford, D. and Sterman, J., Modeling Dynamic Development Processes. System Dynamics Review. 14:1:31-68. Spring, 1998 (b).

Forrester, J., Industrial Dynamics. Productivity Press. Portland, OR. 1961.

Melsa, J. L. and Smith, D. P., Systems and Software Development Leadership: A Practical Improvement Approach. Systems Engineering. 1:1 (May):44-56. 1998.

Richardson, G. P. and Pugh, A III., Introduction to System Dynamics with Dynamo. Productivity Press. Portland, OR. 1981.

The Common Denominator

Len Karas, leonard.karas@lmco.com, Donna Rhodes, donna.rhodes@lmco.com

Abstract. This article discusses the latest enhancement to the systems engineering methodology in use at Lockheed Martin Federal Systems (LMFS) in Owego, New York. The LMFS-Owego methodology has evolved over a fifteen-year period, through evaluation, selection, and synthesis of many successful techniques used for systems analysis and design. The most recent enhancement to the methodology is the *Operational Description Template,* which serves as a **common denominator** for optimal systems development. The insertion of this construct in the existing methodology has been focused on enabling a multi-discipline team to understand and specify how the system will operate, not just the functions it will perform. This becomes the basis for the entire team's work products, providing an effective means for communication with the customer and the various multi-discipline team members, and serving as the basis for cost estimation.

Background

Lockheed Martin Federal Systems (LMFS) in Owego, New York is a premier provider of total systems integration solutions for defense and commercial customers world-wide with 3,800 employees in the U.S. and the U.K. Our diversified business base includes more than 60 programs in the key business areas of aerospace systems, postal systems, and information systems. With our diversified business and the complexity of the solutions we develop, the management of process and technology is critical to success. LMFS-Owego has had a very successful track record over the past years. We achieved a very aggressive plan for business diversification, while significantly increasing our revenue. LMFS-Owego is rated ***Level 5*** by the Software Engineering Institute, and is currently assessing systems engineering practices using EIA IS/731.

The LMFS-Owego methodology was initiated in the early 1980s. At this time, multiple ad-hoc processes and informal methods were being used in different phases of a program, with rough transitions between phases. The result was duplication of effort, technical confusion, and rework. To address this problem, we established a structured incremental methodology (Karas & Rhodes, 1987), applied at each phase of development, using three views of the system (operational, functional, and physical). Formal peer reviews and inspections were also used to promote quality in the design process.

In the late 1980s, analysis showed many development problems to be rooted in lack of time for the operational view, with engineers spending about 90% of time on the vertical view and 10% on operational needs, when the reverse was what we felt was needed. Through modification of the methodology and use of enabling toolsets, we increased the emphasis on the operational (or behavioral) model of the system (Karas & Rhodes, 1993).

As we moved into the 1990s, further process analysis indicated development problems related to the fact that the methodology was difficult to implement fully with the tools and techniques at hand. We lacked the common framework to integrate the system views and to serve as a common reference for all disciplines. Additionally, we lacked formal techniques for optimizing the solution for each dollar spent by the customer. To address these challenges, we inserted a formal construct to provide this missing common denominator, the Operational Description Template (ODT). The ODT was developed to address several difficulties in the methodology/toolset used on previous programs:

1. Graphical representations of behavior diagrams were difficult to generate. Our experience was that the team needed continuous support to ensure groundrules were followed and diagrams were correctly and consistently produced.
2. Our customers could not easily understand the diagrams generated; they first had to be educated in how to read the various diagramming constructs. Customers showed rapid, high acceptance of the ODT.
3. We were unable to represent effectively (and present) the three view system model in an integrated way. Multiple constructs were needed to communicate the information in a single ODT.
4. Team productivity in developing the ODTs proved to be much higher than with the traditional behavior diagram representation, and less support was required to maintain these.

The ODT is a construct designed to enable a multi-discipline team (including customer) to collaborate effectively in understanding, analyzing, and specifying how the system will operate. We have found the tabular format of the ODT to be the most effective means to communicate this information in complex systems. Early applications of the methodology show great promise in optimizing the systems solution and providing the customer with maximum value for each dollar spent.

Process Overview

The LMFS-Owego development process transforms the user need to an operational system, through an iterative process of specification and design. The process is applicable for any program, project, study contract, or internal effort that involves system development. Throughout the process, our systems methodology is applied in each of the phases, and refines the information from the previous phase to lower levels of detail. In each phase the applicable work products are completed, and preliminary work products are generated for the next phase. All members of

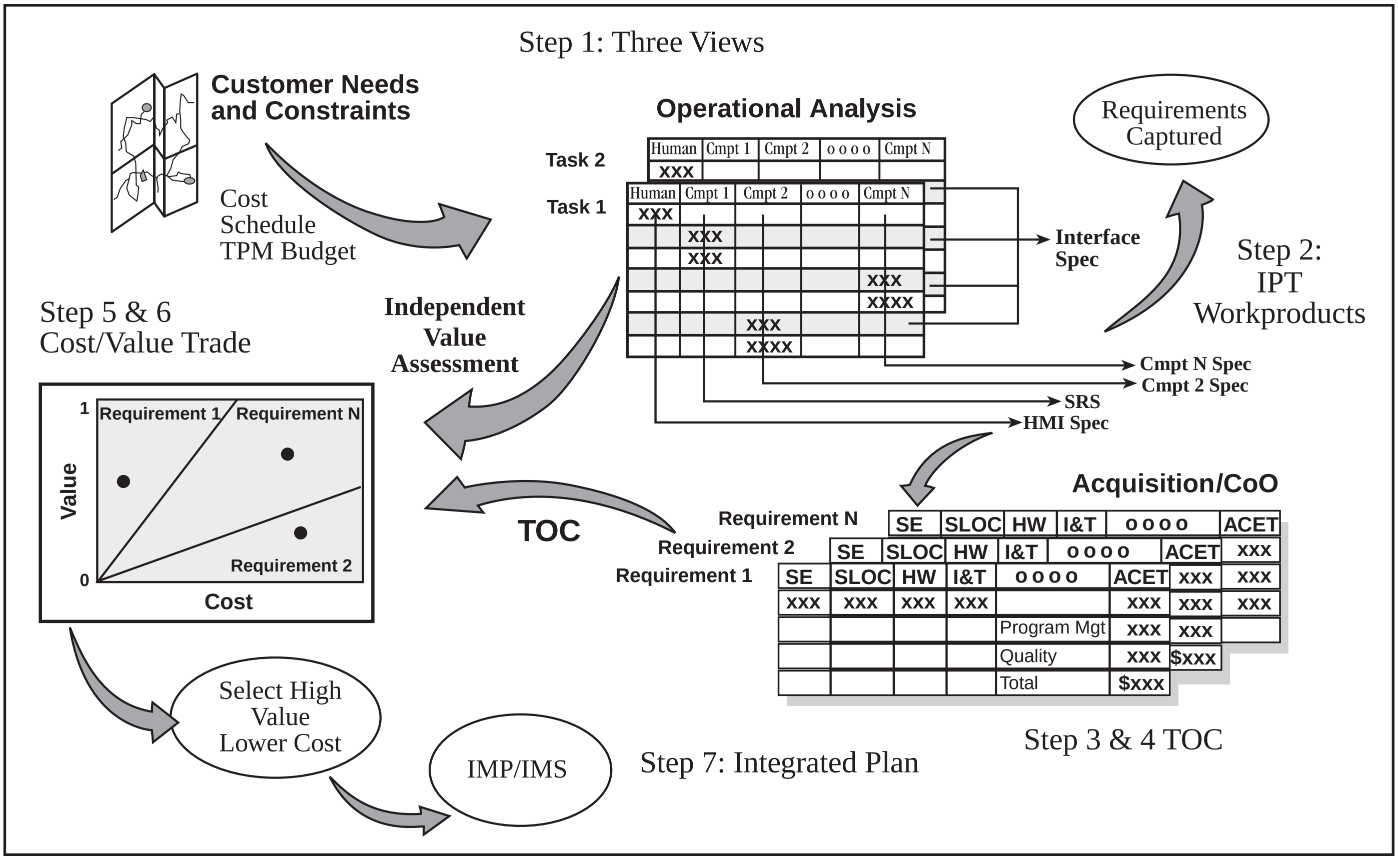


Figure 1. System Development Methodology

an Integrated Product Team (IPT) begin working on their work products in the Requirements Analysis and Planning Phase and continue to refine them throughout each phase. Cost is assessed in each phase.

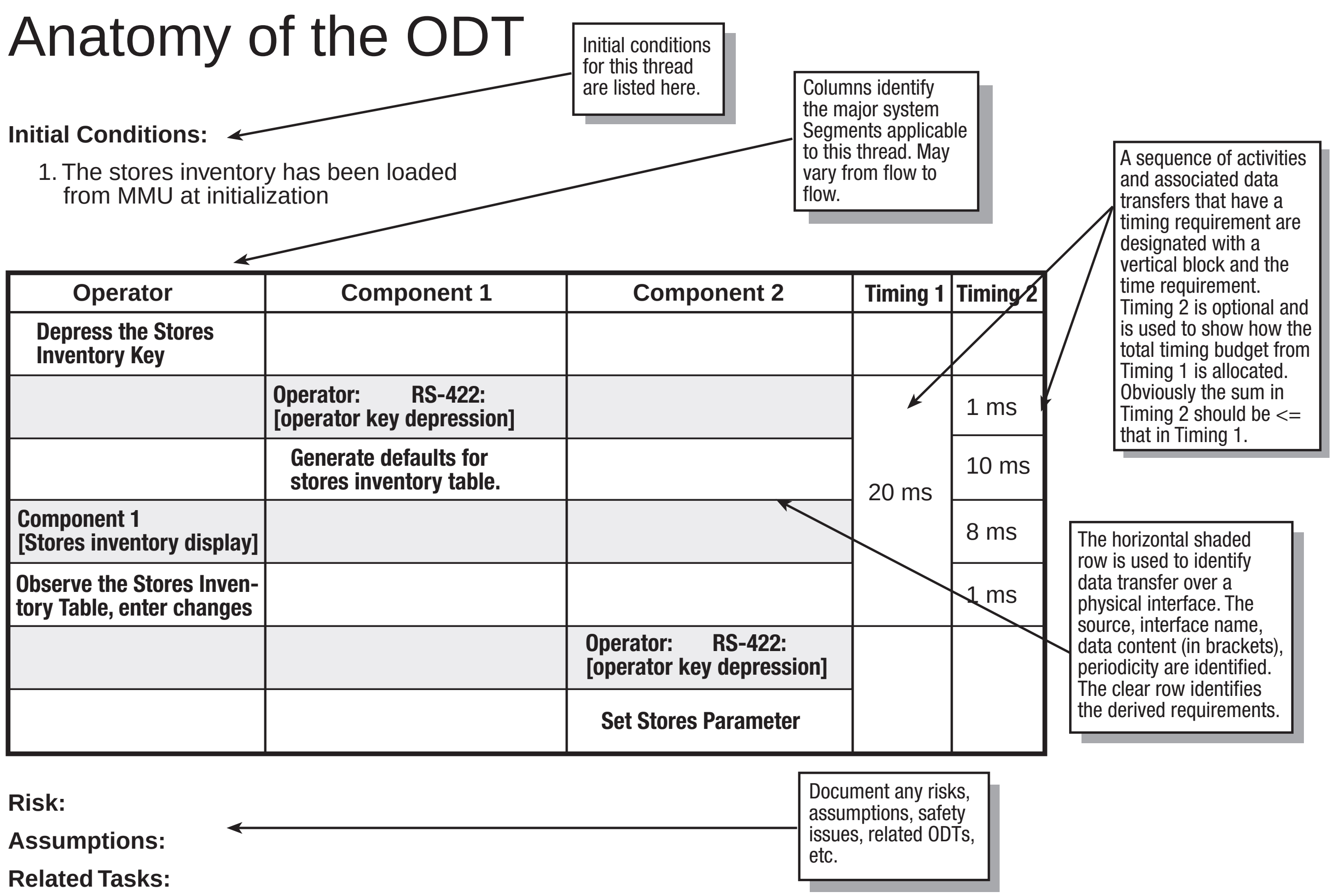
In each life cycle phase, the methodology encompasses the development of three system views (Karas & Rhodes, 1987). The three views are the Functional, Physical and Operational. The Functional View is used to refine the customer's requirements into functions that make the system definition and development manageable. The Physical View represents the physical architecture of the system, and is used to determine if there is a feasible design that can be used to implement the requirements identified in the functional view. To be feasible the design must be affordable, testable, maintainable, reliable, and technically possible. The Operational View maps the functions to the physical design in a time ordered sequence (threads) to model system performance and maintainability. Interfaces between physical components are validated and the data transferred over the interfaces is identified. The Operational View determines suitability. To be suitable the solution must meet operational requirements, performance requirements, and be useable. There are seven key steps in the systems development methodology, as shown in Figure 1.

We will now describe the seven steps in the methodology.

Step 1 — Develop the Three System Views. The key objective in the Requirements Analysis and Planning phase is to determine if the intended scope is achievable within technical and cost constraints. Step 1 begins with requirements from an Operational Requirements Document. The IPT will create a list of system operational and maintenance tasks (as applicable) necessary for the system to satisfy the requirements. The IPT then creates operational and maintenance descriptions for each task using a simple but effective ODT.

Using the ODT template the three system views are developed. The columns represent the Physical View (Architecture). Shaded rows represent physical interfaces between system components. Within each column, functions (Functional View) are identified and allocated to the component. Data is identified and recorded in a shaded cell within a column along with its source. Placing the functions together in a time-ordered sequence provides an operational description or thread for a given system task. All operational requirements are recorded using this technique, resulting in a set of descriptions that characterize the system operation and performance. These ODTs are reviewed with the customer and when approved, form the basis for the IPT work products. Review exit criteria include: suitability, feasibility, ability to meet operational/performance requirements, usability, affordability, maintainability, reliability, testability.

Step 2 — Extract the IPT Work Products. Once the operational and maintenance descriptions are reviewed, the IPT work products are



easily derived from the ODT as shown in Figure 1.

Step 3 — Develop the Cost Estimate for the Total Ownership Cost (TOC). Once options are defined, a solid requirements base is available for cost estimation. We continually collect metrics to use as a Basis of Estimation (BOE). The design information is input into estimators to obtain an accurate cost estimate. Estimators are constantly updated as metrics are collected. Once cost estimates are available, total cost for implementing the requirements can be determined.

Step 4 — Compare the TOC Estimate to the Cost Target. If the total cost exceeds the customer cost constraint, the first step is to revisit the result developed in Step 1. Analysis is done to see if the preliminary design is too far up on the performance/cost curve. A refinement is made to achieve the performance level that is appropriate to meet the need. The cost is then estimated and compared to the cost constraint. If

Workproduct	Derivation
Software Requirements Specifications	Requirements are extracted from each clear cell in the columns with software impact. Data requirements are derived from shaded cells in the columns. This technique is called "walking the column."
Hardware Specifications	Requirements are extracted from each clear cell in the columns that have hardware impact. Data requirements are derived from the shaded cells in the columns.
Human Factors/ Training Requirements	Each Crew column from all operational descriptions are walked and the Human Factors and Training requirements are derived from the clear cells.
Maintainability Requirements	Each Crew column that has maintenance crew impact from all operational descriptions are walked and the maintainability requirements are derived from the clear cells.
Interface Control Documents	The shaded rows are used to capture the physical interfaces and the data requirements. Each shaded cell that has information listed will indicate the source and data that is sent. The information is pulled from all of the operational descriptions and is used to create the ICD.
Integration and Test Plans and Procedures	Each operational description will describe a stimulus/response sequence that the system must perform. This is also a test procedure to verify the system meets requirements. Test planning can be done early and test procedures continually built/refined as the design proceeds.

Table 1. Workproduct Extraction

the cost constraint is still exceeded, the Cost/Value assessment is performed.

Step 5 — Develop an Independent Value Assessment. If the cost estimate for all requirements exceeds available funds, a trade is needed to determine what requirements may need to be deferred. A value assessment is made for each requirement set under consideration by doing a pair-wise comparison of each against all others. The pair-wise approach (Karlsson, J. & Ryan, K., 1997) minimizes subjectivity that is characteristic of doing an ordered list ranking. Once all comparisons are made, a consistency ratio is calculated to determine if any comparison errors were made. After any errors are corrected, a relative order of value is computed for each option.

Step 6 — Perform the Cost/Value Trade. Value and cost are brought together on the COST/VALUE graph, as shown in Figure 1. Each option is plotted according to its value and cost. Those options in a region that have a Value/Cost ratio of 2 or greater are considered high value, low cost and are desirable. Those options in a region that have a Value/Cost ratio of 0.5 or less are considered low value, high cost and are candidates for deferral. This technique allows selections that maximize value for available funds.

Step 7 — Develop/Refine Integrated Master Plan/Integrated Master Schedule. Once the options are selected, the corresponding Work Breakdown Structure (WBS) is created, and used to derive the IMP and IMS. Again a solid base of information is available to determine Events, Significant Accomplishments, and Accomplishment Criteria. Tasks are derived to create an IMS. Metrics are used to estimate the duration for each task.

The methodology described is used in all subsequent phases of the systems development process. Cost is continually monitored through each phase, and if thresholds are exceeded, the Cost/Value trade can be performed to redefine the scope. We now describe some initial results on the application of this methodology.

Initial Results

We have applied the ODT to front-end program activities and achieved excellent outcomes. The following example discusses how we used this process to help determine the scope for a program that would enhance the capabilities of a major avionics system. The team consisted of six systems engineers, two test engineers, two software engineers and customer. Four hundred customer requirements were traced to a design that added operator assist and mission-planning functions, and enhanced mission displays to a major avionics system.

The team spent 90% of the time developing the system behaviors using the ODT. During a four-week period, 80 operational flows were developed and reviewed with the customer. The integrated team was able to identify testability, performance and usability issues early, and resolve them on paper. Hardware and software specifications were then generated from the ODTs. These specifications were not in their final form, but instead, listed the requirements that were derived from the ODTs. These specifications were adequate to get vendor quotations to complete a proposal for this phase of the program. Our team produced these specifications in half the time that is normally needed using a non-operationally based process.

As the effort progressed, it became evident that the customer had more requirements than dollars. Because we had thorough system behavioral descriptions that the customer helped create, and cost data that was based on past metrics, there was no debate when the cost was determined to be twice the available budget. Using a pair-wise comparison cost value technique, our customer assigned relative values to the requirements. Once married with our cost estimates on a cost-value graph, the customer was able to optimize the value for available dollars, and reduce his program from $250M to $100M with rationale for the deferred requirements.

Within a six-week period, we were able to develop a conceptual design, determine the cost, and rescope the program to achieve the best value for the available budget. We learned the following during this effort:

1. Most of the team's effort should be focused on developing the behavioral descriptions. Ninety percent on the ODT completion and ten percent in the specifications seems to be a reasonable allocation.
2. The ODT method is easy to learn and easy to use during reviews, thus, even an untrained team can become extremely productive in a very short time. The team remains focused on solving system issues.
3. Allow the customer to be an integral part of the process, if possible. If the customer has contributed to the behavioral descriptions, is familiar with your cost metrics, understands how you have applied the metrics, and has applied his value assessment, the IPT's results will be optimal.
4. The ODT can effectively serve as the common denominator for the multi-discipline team. All members contribute to development efforts within the single framework and were able to produce their discipline specific work products from the ODT.

Current Efforts

LMFS-Owego has an aggressive effort in place to develop the "production" version of the ODT construct, with an underlying supporting tool environment. We are enhancing our formal procedures and educational materials to permit widespread insertion of the methodology enhancement. We have begun to work with tool vendors to fully automate the ODT, and we hope to encourage other software vendors to work with us. As we complete this effort, plans are to transfer this methodology and supporting tool across Lockheed Martin as part of our LM21 Best Practice Transfer Program. As with all methodologies, the discriminating factor is how well the team applies it, rather than the method itself. Effective use of the ODT is made possible with supporting automation, and we hope to encourage widespread support for this construct. We also hope our systems engineering colleagues will review our efforts and provide useful feedback. We intend to publish the results of full application of the ODT-based methodology in the INCOSE 2001 symposium. In addition to using the ODT on LMFS Owego programs, we have also begun to use the ODT internally for modeling process and planned technology infrastructure improvements.

Summary

We have evolved our methodology over a 15-year period of process improvement. Each evolution of the methodology has produced unique improvement results. The cornerstone of the most recent methodology enhancement is the Operational Description Template. The operational based approach focuses the team on the system behavior rather than functions, quickly providing a conceptual solution to meet customer requirements. The outcome of the operational analysis is captured in the ODT, the common denominator for the entire team. This conceptual solution baseline provides the basis for cost estimation.

Biographies:

Len Karas *is a Senior Systems Engineer in Aerospace Systems Engineering at Lockheed Martin Federal Systems in Owego, New York.*

Donna Rhodes *is Senior Manager of Systems Engineering Technology at Lockheed Martin Federal Systems in Owego, New York, and President Elect of INCOSE.*

References:

Karas, L. & Rhodes, D., "Systems engineering technique," NATO Advisory Group for Aerospace Research & Development Conference Proceedings No. 417, December 1987.

Karas, L. & Rhodes, D., "Enabling concurrent engineering through behavioral analysis," Proceedings of the Third International Council on Systems Engineering, July 1993.

Karlsson, J. & Ryan, K.,"A cost-value approach for prioritizing requirements," IEEE Software, 1997, 67-74.

Development Integration and Implementation of Engineering Processes at Oerlikon Aerospace

Claude Y. Laporte, claporte@yortar.com, Sylvie Trudel, strudel@oerlikon.ca

Abstract: In order to reduce cycle time, increase customer satisfaction and lower costs, Oerlikon Aerospace initiated a series of projects to define and implement engineering and management processes. The first initiative, in 1992, defined a software engineering process. A second initiative was started in 1995 with the objective of defining and implementing a systems engineering process, and integrating this process to the software engineering process already in use. We present a brief description of the context, then describe the systems engineering process. Organizational mechanisms to better manage changes are also described. Finally, lessons learned are presented.

Process Development Background

Oerlikon Aerospace (OA) is the systems integrator of an air defense missile system. More than 100 systems and software engineers were involved in the development and maintenance of the system. In fall 1992, recognizing that engineering was a core competency, the OA president approved the budget for a software capability assessment, as well as for the preparation of a Process Improvement Plan (PIP). In spring of 1993, assessors certified by the Software Engineering Institute (SEI) performed a formal software assessment. During a second formal assessment conducted in February 1997, OA achieved a strong SEI level 2 certification, and even satisfied eight of seventeen goals for SEI level 3 certification.

Although the organization had been ISO 9001 certified since 1993, it was decided that a systems engineering process also had to be developed in order to seamlessly integrate disciplines associated with systems engineering. In 1995, a mini assessment of systems engineering practices was performed. After the assessment, it was decided to use, as frameworks, the Systems Engineering Capability Maturity Model (SE-CMM)©[1] and the Generic Systems Engineering Process (GSEP)©[2] developed by the Software Productivity Consortium (SPC 1995). An in-depth description of the systems engineering process has been presented at a symposium of INCOSE (Laporte 1997).

Development of a Systems Engineering Process

The GSEP document describes, using the IDEF notation (USAF 1981), management and technical activities, and the artifacts produced by each activity. The major management activities are: Understand context, analyze risk, plan increment development, track increment development and develop system. The major technical activities, as illustrated in Figure 1, are: Analyze needs, define requirements, define functional architecture, synthesize allocated architecture, evaluate alternatives, validate and verify solution, and control technical baseline. Each

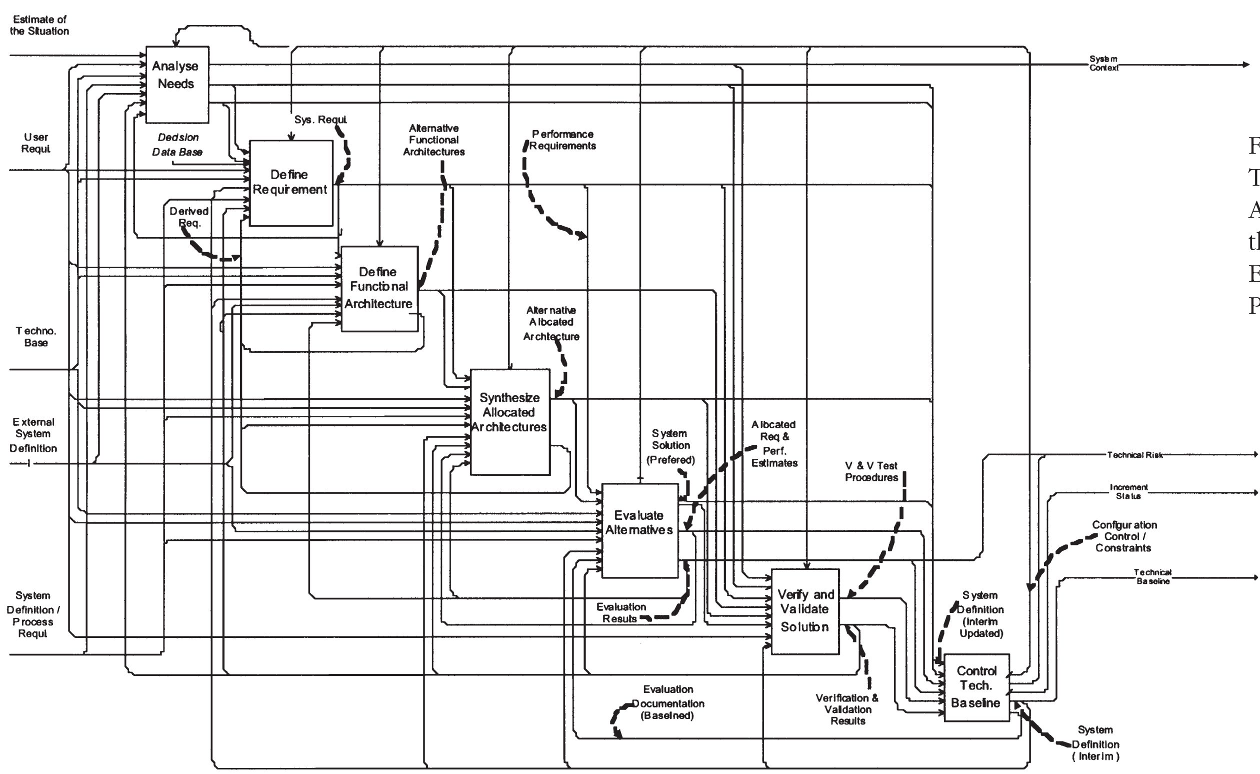


Figure 1: Technical Activities of the Systems Engineering Process

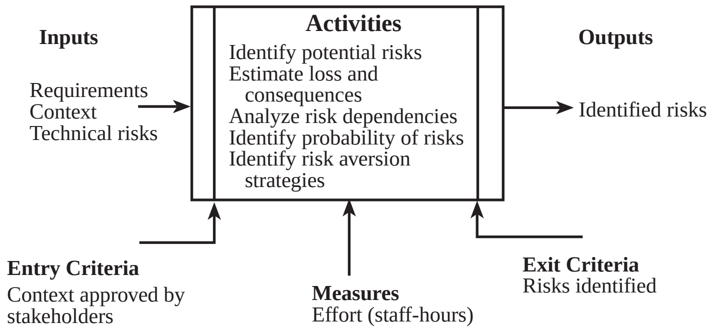


Figure 2. Example of a modified ETVX Notation – Perform Risk Analysis

major activity is broken down in a certain number of smaller activities that are described, individually using a modified Entry criteria-Task-Validation-eXit criteria (ETVX) notation (Radice 85). This notation was also used to document software process and management processes such as the project management process. As an example, the "Analyze Risk" top-level activity is composed of four lower level steps: Perform Risk Analysis, Review Risk Analysis, Plan Risk Aversion, and Commit to Strategies. One step titled Perform Risk Analysis is illustrated, using the modified ETVX notation, in Figure 2.

Integration of the Software Engineering Process to the Systems Engineering Process.
We have used, as a framework to integrate the software engineering process to the systems engineering process, a document produced by the SPC entitled: Integrated Systems and Software Engineering Process (ISSEP)©[3] (SPC 1996). ISSEP defines a set of management and technical activities and the following interfaces: (1) interfaces between the management and technical activities, (2) interfaces among management activities, (3) interfaces among technical activities, and (4) interfaces between the systems and software or hardware development processes. Similarly to the GSEP, ISSEP is adaptable and tailorable to a range of applications and project environments.

Deployment of the Systems Engineering Process
The systems engineering process was deployed for the first time for the re-engineering of two subsystems of the air defense system, namely the launcher control electronics and the operator consoles. The launcher control subsystem is composed of a main data processor which coordinates the operation of the sensors, the launch and guidance of the missiles; a missile tracker processor; a target tracker processor; and a servo control processor. The operator consoles consist in a radar console, which allows controlling the radar and communication subsystems, and an electro-optical console, which allows controlling optical sensors and missile launcher.

Both re-engineering projects were divided into increments: a definition phase and a detailed hardware/software development phase. The identification of each increment was based on the nature of the deliverable product at the end of the increment. In both cases, the first increment deliverable was a system requirement specification, and the second increment deliverable was a set of design and equipment specifications, plus a qualified working pre-production prototype. An in-depth description of the re-engineering project can be found in paper presented at the 1998 INCOSE Symposium (Laporte 1998).

The Management of Change
Since the management of change is a key element of a successful process improvement program, a series of actions were planned in order to facilitate the development, the implementation and the adoption of the processes, methods and tools. As an example, to build the sponsorship level, the president attended a one-day executive seminar on process improvement and two directors attended a three-day seminar discussing the CMM, process, process assessment and improvement. Briefing sessions were held and articles were written in each company's newsletter to explain the why, what and how of process assessment and improvement activities and describing the progress made. Finally, surveys were conducted to assess the organization's readiness to such a change in practices. The surveys identified strengths of the organization and potential barriers to the planned improvement program.

Also, in order to get support and commitment for the future implementation of processes, working groups were staffed with representatives from many departments, including software engineering, systems engineering, sub-systems engineering, quality assurance, contract management, and configuration management. Each working group was managed like a project. It had a charter, a budget and a schedule. A process owner, (i.e. a manager responsible for the definition, implementation and improvement of each process) was part of the working group. A member of the working group acted as a facilitator in each group. Therefore, the process owner would focus on the content of a specific engineering process while the facilitator would focus on the process of developing a process.

Lessons Learned
Certain lessons that could benefit other organisations in the future are discussed below.

Lesson 1: Tie Process Improvement Activities to Business Objectives. It was observed that software and systems engineering process improvement really picked up momentum when a common focal point was created between management, engineers and customers. They understood that the real benefit of process improvement is that it has the potential to improve product quality, reduce time to market, and reduce cost. Consequently, it improves the ability of an organization compete. Additionally, a multi-year Process Improvement Plan (PIP) was a very important tool to illustrate the links between business objectives, project requirements and process development or improvement. Essentially the PIP illustrated that the engineering of processes was not a paper exercise but an important infrastructure for the successful accomplishment of projects. Being a multi-year plan, the PIP also showed practitioners the long-term commitment of management to business and process improvement activities.

Lesson 2: Train all Users of the Processes, Methods and Tools. Once processes are defined, it is essential to train all users. Otherwise, process documents will end up getting dusty on shelves. It is illusory to think that developers will study, on their own initiative, new processes in addition to their workload. Training sessions also serve as a message that the organisation is moving ahead and will require that its developers use these practices. During the training sessions, it is necessary to indicate that, even with everybody's good will, errors are bound to happen while using new practices. This message may help reducing developers' level of stress when using these new practices. It would be a good thing to have a resource person available to help developers (e.g. on a hot line) when they face obstacles while implementing new practices.

Lesson 3: Manage the Human Dimension of the Process Improvement Effort. We wish to make you aware of the importance of the human dimension in a process improvement program. The people responsible for these changes are often extremely talented engineering practitioners, who may not be trained in change management skills. The reason for this is simple: their academic training focused on the technical dimension and not on the human aspect. However, the major difficulty of an improvement program is precisely the human dimension.

While preparing the technical part of the improvement action plan, the change management elements have to be planned. This implies, among other things, a knowledge of (1) the organisation's history with regard to any similar earlier efforts, successful or not; (2) the company's culture; (3) the motivation factors; and (4) the degree of urgency perceived and communicated by management, the organization's vision, and genuine support. We are convinced that the success or the failure of an improvement program has more to do with managing the human aspect than managing the technical aspect.

Lesson 4: Process Improvement Requires Additional "People Skills." In an organisation that truly wants to make substantial gain in productivity and quality, a cultural shift will have to be managed. Such a cultural shift requires a special set of "people" skills. The profile of the ideal process facilitator is someone with a major in social work and a minor in engineering. The implementation of processes implies that both management and employees will have to change their behaviour. With the implementation of processes, management will need to change from a "command and control" mode to a more "hands-off" or participative mode. As an example, if the organisation truly wants to improve its processes, ideas should come from those who are working, on a daily basis, with the processes. This implies that management will need to encourage and listen to new ideas. This also implies that the decision making process may have to change from the autocratic style, e.g. "do what you are told," to a participative style, e.g. "let us talk about this idea." Such a change requires support and coaching from someone outside the functional authority of the managers who have to change behavior. Similarly, employees' behavior should change from being the technical "heroes" that can solve any problem, to team members that can generate and listen to others' ideas.

Facilitating behaviour changes requires skills that are not taught in technical courses. It is highly recommended that the people responsible for facilitating change be given appropriate training. The authors recommend two books that may facilitate the management of change: the first one (Block 1981) gives advice to anybody acting as internal consultant; the second one (Bridges 1991) provides the steps to be followed for writing and implementing a change management plan.

Lesson 5: Select Pilot Projects Carefully. It is also very important to select carefully pilot projects and participants to the pilots since these projects will foster adoption of new practices throughout the organization. Also, first time users of a new process will make mistakes. It is therefore mandatory to coach properly the participants and provide them with a "safety net." If participants sense that mistakes will be used to learn and make improvements to the process instead of to "point fingers," the level of anxiety will be reduced and they will bring forward suggestions instead of "hiding" mistakes. As an example, the main objective of a formal inspection process is to detect and correct errors as soon as possible in the project lifecycle. Management has to accept that in order to increase the errors detection rate, they should not make public the results of individual inspection, but only the composite results of many inspections. When management accepts this rule, employees may feel safe to identify mistakes in front of their peers instead of hiding them. The added benefit to correcting errors is that those who participate in an inspection may learn how to avoid these errors in their own work.

Lesson 6: Conduct Process Audits. Process audits should be conducted on a regular basis for two main reasons: First, to verify that practitioners are using the process, and second, to discover errors, omissions, or misunderstandings in the application of the process. Process audits help to assess the degree of utilization and understanding by the

Activity	Results from First Audit	Results from Second Audit
Comments made by reviewers	38 %	78%
Approval matrix completed	24%	67%
Effort log completed	18%	33%
Review checklist completed	5%	44%
Configuration management checklist completed	5%	27%
Distribution list completed	38%	39%
Document formally approved	100%	100%

Table 1. Results of audits performed on the Documentation Management Process

practitioners. As an example, a documentation management process was released and practitioners were asked to produce and update documents using this new process. It is widely known that engineers are not prone to documenting their work. An audit was launched to measure process compliance. As expected (see Table 1), results of the first audit were not exhilarating. The engineering manager kindly reminded engineers, in writing, to use the process. He also informed them that a second audit would be performed. As shown in the table, the results of the second audit are substantially better than the first audit. Also, the auditor gathered feedback from engineers; this information is used by the process owner to improve the process.

Lesson 7: Conduct Team Effectiveness Surveys. Surveys (Alexander 1991) may promote open discussion with members of a group since most people are not inclined to raise "soft" issues. Also, such tools provide the facilitators with information that help them probe delicate issues. As an example, if the majority of a working group reports that interpersonal communications are weak, the facilitator can probe the members and invite them to propose solutions. After a few meetings, the results of a new survey will show if the proposed solutions really helped the team improve performance and communication.

Lesson 8: Get Support from Organizational Change Experts. As mentioned above, surveys were conducted in order to "measure" issues such as culture, implementation history, and team effectiveness. Once the surveys were compiled, we had some indications of organizational strengths and weaknesses. The difficult part was to decide what to do next. As an example, one issue from the survey is taking risks, and that people are not willing to take risks. One possible reason for such behavior was that people did not want to be blamed for an error. Having found this cause was not too helpful, since we would have no influence over the cause for this behavior. It would have been very helpful to have access to someone with expertise in organizational change. This would have saved a lot of long discussions and many wrong answers.

Lesson 9: Start a Process Initiative from the Top Level Process. The process improvement initiative was a bottom-up exercise, i.e. first software process was developed, then systems engineering process, then project management process where. Each additional process "sits" on top of the other. Historically, this was the selected strategy because, in 1992, only the software CMM was available; then, came the systems engineering CMM and after, the Body of Knowledge in project management (PMI 1996). If an organization had to start a process initiative today, it would be easier and more efficient to start from the top by developing the project management process, then the systems engineering process and finally the software process. It would also be possible to develop these processes in parallel once the requirements for the top-level process are stabilized.

Lesson 10: Adopt a Common Vocabulary. To succeed in any project endeavor, a common vocabulary is a basic requirement. As we developed these processes, we realized that different players had different meaning for the same word, or the same word had different meanings, and some words were not well known to some individuals. We therefore mandated one team member as the "glossary keeper." His role was to collect a vocabulary, propose some "clean-up" in the terminology, and to build gradually a common glossary for all processes.

Conclusion

We have shown that the development and deployment of engineering and management processes entail technical and management competencies. Five elements are necessary for successful implementation of organizational changes. First, management sets a direction, and process objectives are linked to business objectives. Second, people are trained to perform new tasks. Third, incentives are provided to facilitate the adoption of changes. Fourth, resources are estimated and provided. Fifth, an action plan is developed and implemented. We also learned that the constant attention to the "people issues" is critical to the success of a change project.

Improvements required significant investments, but both the technical and management processes will allow complex projects to be developed in a disciplined environment.

As a final word, a quotation from Pfeffer: "It is almost impossible to earn above-normal, exceptional economic returns by doing what everyone else is doing. It is also impossible to achieve some lasting competitive advantage simply by making purchases in the open market – something that anyone can do." (Pfeffer 98).

Footnotes:

1 SE-CMM is a service mark of Carnegie Mellon University

2 Copyright by the Software Productivity Consortium

3 Copyright by the Software Productivity Consortium

References

Alexander, M., The Encyclopedia of Team-Development Activities, edited by J. William Pfeiffer, University Associates, San Diego, California., 1991

Bate, R., "A Systems Engineering Capability Maturity Model," version 1.1, Software Engineering Institute, CMU/SEI-95-01, November 1995.

Block, P., Flawless Consulting, Pfeiffer & Company, 1981.

Bridges, W., Managing Transitions, Addison Wesley, 1991.

EIA, "Systems Engineering Capability," EIA/IS 731.1 Part 1: Model, Electronic Industries Alliance, 1999.

Laporte, C., Papiccio, N., Development and Integration of Engineering Processes at Oerlikon Aerospace, Proceedings of the Seventh International Symposium of the International Council on Systems Engineering, August 3-7, Los Angeles, California, 1997.

Laporte, C., Guay, A., Tousignant, J., The Application of a Systems Engineering Process to the Re-Engineering of an Air Defense System, Proceedings of the 8th International Symposium of the International Council on Systems Engineering, July 26-30, Vancouver, Canada, 1998.

Pfeffer, J., "The Human Equation Building Profits by Putting People First," Harvard Business School Press, 1998.

Paulk, M. et al, "Capability Maturity Model for Software," Software Engineering Institute, SEI/CMU-93-TR-24, 1993.

PMI, "A Guide to the Project Management Body of Knowledge," Project Management Institute, 1996.

Radice, R., "A Programming Process Architecture," IBM Systems Journal, vol. 24, no. 2, 1985.

SPC, "A Tailorable Process for Systems Engineering," Software Productivity Consortium, SPC-94095-CMC, January 1995.

SPC, "Integrated Systems and Software Engineering Process," Software Productivity Consortium, SPC-96001-CMC, May 1996.

USAF, "Integrated Computer-Aided Manufacturing Architecture," Function Modeling Manual (IDEF0), United States Air Force, AFWAL-TR-81-4023, 1981.

Biographies

Claude Y. Laporte *has a Bachelor in Science, a MS in physics, a MS in Applied Sciences. He was an officer in the Canadian Armed Forces for 25 years. At Oerlikon Aerospace he coordinated the development and implementation of engineering and management processes. In 1999 he started to offer consultating services in process engineering and change management.*

Sylvie Trudel *obtained in 1986 a Bachelor degree in Computer Science. She worked 10 years in development and implementation of management information systems. She joined Oerlikon Aerospace in 1996 as the process control analyst.*

Working Groups

Tools Integration and Interoperability Working Group

John Nallon, nallon@tdtech.com

The Tools Integration and Interoperability Working Group (TIIWG) is developing operational scenarios and requirements for an integrated systems engineering environment. This entails every aspect of an environment from process to tool functionality, including operational integration and tool communication. The TIIWG is a working group of the Modeling and Tools Technical Committee chaired by Mark Sampson (TD Technologies).

This article is an update of our recent activities and the preparation for the International Workshop in January. The TIIWG met five times during the International Symposium in Brighton, and we were pleased to have high attendance at the meetings. It was obvious that the integration of the systems engineering environment holds a high interest with the INCOSE membership.
TIIWG Events at Brighton:

- H04: AP233 Tutorial by Dr. Julian Johnson
- TIWG Working Meeting
- MTTC Dinner: Donatello's Italian Restaurant in "The Lanes"
- Modeling & Tools Technical Committee Open House
- TIIWG Dinner: The Carriage House on the Channel
- Planning Meeting from Now to International Workshop
- AP233 Liaison Committee Discussion

The TIIWG activities at the symposium centered on the AP233 Tutorial, and on designating a representative to the ISO AP233 committee to represent INCOSE during the development of this standard. AP233 is the Systems Engineering Data Exchange Standard for system engineering tools. The standard is very important to the development of tools to support an integrated environment. The AP233 standard will enable all tools to integrate with a common information model and data format. We also discussed the continuing development of the TIIWG products that are currently in development. Our current plans are focused on actions to complete the first draft in time for the International Workshop. This and other action items will be disseminated in the next couple of months.

Integrated System Engineering Environment

Co-chair Jim Schier (TASC) reviewed a method of organizing the TIWG products. We are planning to follow the approach that maps EIA-632, EIA-731 and ISO 15288 into our work products. Some enterprise processes exert significant impact on our engineering environments and we concluded that these processes are outside the scope of 632. To address this issue, we added four technical volumes to address the Enterprise Processes identified in ISO 15288, the emerging standard on Life Cycle Processes. More detailed information on our products and work to date can be found on the INCOSE Web site at www.incose.org/tiiwg.

AP233 Liaison Selection

The number one action recently addressed was the selection of the INCOSE representative to the ISO AP233 standard committee. Two candidates had been identified prior to the Symposium. The Terms of Reference (job qualifications) were reviewed, prioritized, and approved. Nominations were opened for consideration from a wider pool of people. This resulted in nine potential candidates. Those who nominated candidates were asked to approach their nominees and ascertain their degree of interest and their ability to make the commitment. A tentative schedule for completing the selection process was developed for ratification. After candidate qualifications were reviewed during July, Dr. David Oliver was selected as prime candidate, and Doug Stemm (Raytheon) was selected as alternate. The INCOSE Technical Board ratified this selection in August 1999.

The MWG Helps Practical Software Measurement Get A Systems Perspective

Garry Roedler, Chair, Measurement Working Group (MWG), garry.j.roedler@lmco.com

During the past two years, the INCOSE Measurement Working Group (MWG) has been working with the Practical Software Measurement (PSM) initiative to define the needs for measurement related to the engineering of systems, and to determine how to address those needs in the PSM guidance. PSM is an issue-driven measurement process whose development is sponsored by the Office of the Under Secretary of Defense for Acquisition and Technology. It is based on proven practical experience of government and industry and has become the *de facto* standard for software measurement. Our association with PSM started in 1997 as a study group led by Garry Roedler (Lockheed Martin Management and Data Systems) and Dr. William Farr (Naval Surface Warfare Center). The study group evolved into a collaborative project between the INCOSE MWG and PSM to develop joint measurement products that address the systems engineering needs and meet the following objectives:

- Integrate SE measurement

guidance with the existing PSM approach

- Use the PSM measurement process with specific SE measures, indicators, and analysis
- Span measurement needs for engineering most systems
- Link systems and software analysis
- Ensure applicability of the measurement guidance to new/existing project implementations
- Allow guidance to be developed and refined in a phased implementation

The PSM Process

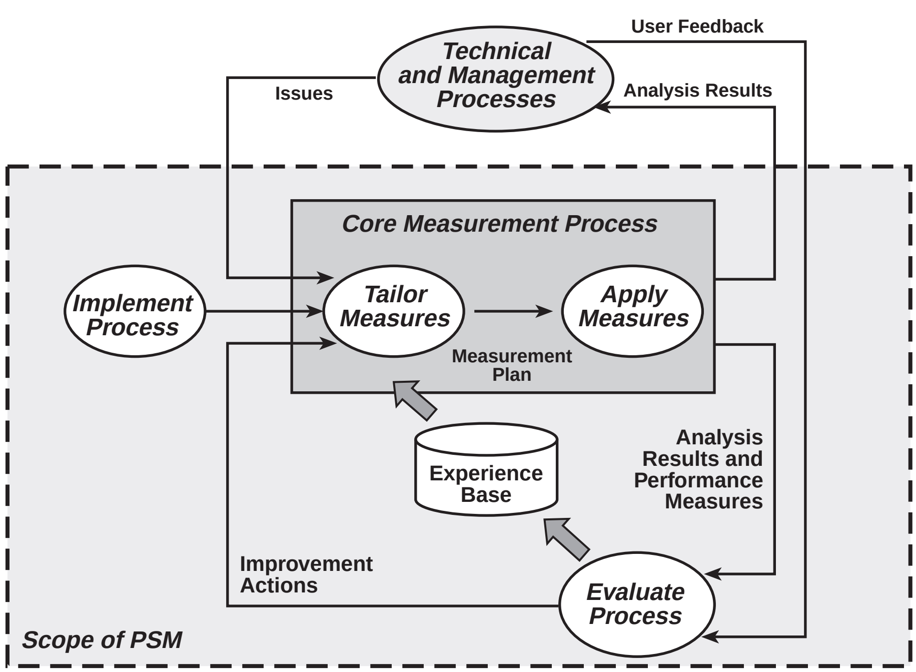
The underlying concept of the PSM process is that measures should not be pre-defined, but should be tailored to the unique issues that exist for each project. The PSM Guide defines nine principles of measurement, the first of which is: "Project issues and objectives drive the measurement requirements." Close scrutiny of these nine principles on which PSM is based showed that all are as valid for systems applications as they are for software. The Guide defines an end-to-end measurement process that includes a three-step activity to derive tailored measures for each project. This three-step activity includes identifying the specific project issues, placing each issue in the appropriate category that is defined in the Guide, and then selecting and specifying appropriate corresponding measures (either from the Guide or from another source). The greatest differences between were found (in systems versus software engineering) in the process used to select the appropriate measures. To provide a complete treatment of the measurement process, PSM also provides in-depth guidance for implementing, applying, and evaluating the measurement process. The Applying Measures activity includes data collection and preparation, analysis, reporting, and use of the measures selected for the project. The Guide includes analysis techniques specifically for estimation, feasibility analysis and performance analysis. The example indicators and case studies that are included assist in this activity. The Implement Process activity focuses on establishing and sustaining the measurement process on the project or within the organization. Finally, the Evaluate Process activity focuses on evaluating performance of and improving the measurement process. This Evaluate Process activity is new to PSM and will be added in the next revision. The following diagram (below) provides the top-level view of the revised process (still in draft form).

Evaluating PSM's Applicability to Systems Measurement

Many customers acquire systems, not just software. However, measurement guidance is usually oriented towards measuring the software characteristics of the system. A measurement process for system development needs to support the entire systems acquisition process. This systems measurement process must integrate proven measurement techniques that have been established for the various engineering disciplines that contribute to a system's development. Evaluation of the PSM measurement process has shown that it meets these systems-oriented needs.

The original goal of the project was to develop a set of guidances for systems engineering that mirrored the guidances available for software. As part of the analysis of the systems engineering measurement requirements, the team investigated the similarities and differences of software and systems measurement. The following figure (see *Comparison of SW and Systems Measurement*, next page) shows the results.

This comparison, along with the current direction of the technical community towards the integration of systems and software engineering guidance and assessment, led us to refocus the project's goal on developing an integrated set of measurement guidance, training, and tools addressing both systems and software project management and engineering. Thus, the new PSM guidance is being written to cover all the measurement needs of systems and software engineering, development, and integration. Accordingly, the new guidance will integrate the existing PSM, INCOSE, and SEI product lines, which provide proven software, systems engineering, and process improvement measurement guidance. As a result of the changes to the goals, the title of the guidance and initiative will be changed to Practical Software and Systems Measurement,



Comparison of SW and Systems Measurement

SIMILARITIES	DIFFERENCES
– **Measurement Process**	– **Primary differences are in measures**
– **Measurement Objectives**	– **System specific measures:**
– **Analysis Techniques**	• Some similarities apply only to systems (e.g. survivability)
– **Most of the same top level issues**	• Recovery measures
• Schedule/Process	• Physical characteristics
• Cost/Resources	• Bill of material measures
• Product Size & Stability	– **Software specific measures:**
• Product Quality	• Software characteristics
• Process Performance	• Functional size (e.g., function points) - no system equivalent
• Technology Effectiveness	
• Customer Satisfaction	– **TPMs are domain/discipline specific**

although the PSM acronym will still be used. The revised guidance will be available by December 1999 and the supporting training and tool updates will be available in the first quarter of 2000.

Changes to PSM Resulting From Integration of Systems

♦ New Common Issue Areas are being added or modified
  - "Customer Satisfaction" is being added as an issue area
  - "Product Size and Stability," "Process Performance," and Technology Effectiveness" are modifications from existing issue areas

♦ Significant changes are being made to the following measurement categories to better focus on systems:
  - Modify the "Physical Size and Stability" category to address physical characteristics of systems components
  - Restructure all of the "Product Quality" categories to better align with current system and software quality standards (to focus on necessary quality factors)
  - Restructure all of the "Process Performance" categories to address more of the facets of process evaluation
  - Modify two of the "Technology Effectiveness" categories to cover more breadth and depth of evaluating alternative or evolving technologies for use
  - Add a "Customer Feedback" category to address the perspectives of the customer in the process

♦ The following new measures are being added specifically for systems:
  - Interfaces
  - Physical Characteristics
  - Recovery Impact
  - System Maintenance
  - Operator Errors
  - Fault Tolerance

♦ Several other measures are being added that apply to both software and systems

For more information, contact a member of the MWG or the PSM Support Center at: http://www.psmsc.com

Measurement: Frequently Asked Questions

Ken Stranc, kjstranc@tasc.com

Question: How do I decide which measures to use?

Selecting the correct metrics is essential for the success of a measurement process. As the practice of engineering measurement has matured, a number of techniques have been developed to support measure definition and selection. Three commonly used measurement techniques include guidance on how to select measures. These are:

1. Practical Software and Systems Measurement (PSM) sponsored by the Office of the Under Secretary of Defense for Acquisition and Technology
2. Goal-Question-Metric (GQM) developed by Victor Basili at the University of Maryland
3. Goal-Driven Software Measurement from the Software Engineering Institute.

You can find references to these documents on the INCOSE MWG web site at http://www.distributive.com/INCOSE_MWG/Home.html.

To begin, you first need to define the information needs that the measures will address at the time you will use them. Start by developing a complete understanding of the objectives of your project. Articulate your project's issues in terms of its problems, risks, and areas where you lack information to support decisions. In developing the list of objectives and issues, consider holding a workshop where your project's stakeholders can voice their interests and concerns.

Once you have compiled a list of all the objectives and issues that you want to address through measurement, it is essential that you prioritize them. Resource constraints and data availability may not allow you to measure everything, so by prioritizing your issues you will keep your measurement activities focused on what is most important to your project right now.

Next, develop a set of questions whose quantitative answers will provide sufficient insight into your project's objectives and issues. Then, using one or more of the guidebooks, handbooks, and tools available, select measures that will provide the required quantitative information to answer these questions and allow you to take action. If no action can imagined which would result from the review of a measure, select a different measure. If no measures will cause an action to be taken, then your objective or issue requires more than a quantitative answer: It requires further development, or perhaps it does not belong on your list at all.

Very importantly, determine how easy or difficult it will be to obtain the required data. Readily available data may be found that addresses a low priority objective or issue which otherwise would not have been addressed. Conversely, data that is difficult to obtain may not be worth the cost, even for a high priority objective or issue that could be better served using a different measure. Also, pay particular attention to matching the granularity of the available data to what is required. For example, if you use labor hours in tenths of an hour but it is only available in whole hour increments, you will need to consider the impact of changing the way labor hours are reported. If you only use labor hours in whole hour increments, collect it that way.

Based upon the priorities of your objectives and issues, and the availability of the data, you can now select the particular measures that you will use. Six to ten measures are generally sufficient to address a project's most critical objectives and issues. Be sure to look for opportunities where you can leverage a measure to address more than one issue.

Before moving on, it is essential that you review and refine the objectives, issues, priorities, and selected measures with the project's stakeholders, as this step ensures their buy-in to the measurement process. In fact, it is always a good idea to review the measurement process with your stakeholders periodically as the project evolves and the issues change. Be particularly aware that issues change as you move from one phase to the next in the project lifecycle.

Question: Given that you can't measure everything of interest at once, how do you prioritize?

Focus on what is most important to your project right now. Your measurement priorities should always align with your project management and systems engineering priorities. What you need to prioritize are not the entities being measured, but rather the objectives and issues for which quantitative data are sought.

After you have identified your project's objectives and issues, you need to rank them according to well-defined criteria that you develop and apply. One simple method, is to separate the issues into three categories based upon their projected impact: definite showstoppers, potential showstoppers, and others. An example of a definite showstopper may be the expenditure of all project resources before the project is completed. An example of a potential showstopper may be failure to meet a particular product size or weight requirement.

If you are quantifying risks, your priorities may align closely with the ranking of your risks. Here you use a weighted calculation of the estimated project impact and probability of occurrence to develop an ordered list of objectives and issues.

It is really up to you to decide what is most important in terms of objectives and issues. Once you have established the set of issues you want to track, select the measures needed to address them. This selection, however, is not solely driven by your prioritization of objectives and issues. Availability of the data may be a factor in the way you prioritize your objectives and issues and in the way you select your measures. For example, you may find that a low priority issue can be addressed very inexpensively by using data whose collection and analysis are already motivated by a much higher priority issue.

Over time, your project's objectives and issues, and their relative priority, will certainly change. It is essential that you review your measurement priorities periodically to ensure that the measures you are using are giving you insight into your most critical concerns.

Liaison between INCOSE and ISO 10303-AP233 Working Group

David Oliver, dwoliver@ix.netcom.com, INCOSE-ISO TC184/SC4 Liaison;
Doug Stemm, djstemm@west.raytheon.com, Backup Liaison;
Sylvain Barbeau, sylvain.barbeau@espace.aerospatiale.fr, AP233-Chair;
Julian Johnson, julian_johnson@bae.co.uk, Technical Manager SEDRES

ISO "STEP" and AP233. Within the ISO Standardization organization, TC184/SC4 is the committee that is developing standards to describe and manage product data throughout the life of the product. It is within that structure that all standardization activities with ISO 10303 "STEP" (amongst other work) is organized. STEP stands for Standard for Technical Exchange of Product data. This work embraces the ISO 1030-AP233 Working Group that is developing an international standard, as a new part of STEP, to capture the information developed in systems engineering in an information model. The base information model has been developed under the ESPRIT program called SEDRES. The work of developing such an information model, the generation of tool interfaces based on the model, and the validation of the effort were described at the 1999 INCOSE symposium in several papers and a tutorial presented by Julian Johnson. In ***INSIGHT*** Vol 2, Issue 2, Sylvain Barbeau summarized AP233 progress.

As a technology, STEP has been in development and use for many years, evolving in the early 1980s out of work with the IGES and SET standards. STEP is large in scope, and can be used to describe anything from an entire process plant, to an individual printed circuit board. STEP covers many aspects of a conventional development lifecycle, providing product data interchange within CAD (computer aided engineering), CAE (computer aided engineering), and CAM (computer aided manufacturing) worlds, collectively known as CAx tools. STEP is being further developed to describe full product life-cycle information; see the information about the NATO Product Life Cycle Support initiative (PLCS) on the International CALS Congress site (http://www.cals-international.org/public/index.htm).

The components of STEP are at different stages of maturity. The portions used to describe product geometry have been in use for a few years. STEP technology is being used for information modeling, for instance, and the data model is expressed using the STEP data modeling language EXPRESS. This standard technology is relevant (1) to the evolution of systems engineering environments with a suite of interconnected tools, (2) to extending the use of models in system analysis and design, and (3) to the emergence of internet based techniques for communicating specifications and designs. Several INCOSE Technical Committees and Working Groups are likely to have interest in these developments.

Status of AP233. Various technical work has been done since the June ISO plenary session held in Lillehammer (see previous issue of ***INSIGHT***). As planned, work has been performed both on the requirements representation and on behavior representation. On requirements, additional features have been added to the model, which will be reviewed during the next meeting in New Orleans (7-12 November 1999). Since several requirement methods are used in the industrial world, the objective is to have a flexible way of describing requirements that is compatible with the various methodologies. The approach adopted enables the core information for a requirement to be associated with the numerous properties that emerge as the system becomes more precisely defined. For the behavior representation, the last release of the data model offered three ways of describing behaviors, the result of the team's effort to create a more unified model.

The document gathering the information on the data exchange data standard is also under revision. In order to disseminate the information with sufficient detail and a correct level of understanding, a "system engineering data exchange requirement" has also been reworked. This document will help people to understand the rationale for using a standard when considering system engineering workbench design in a company.

Finally, several administrative updates have been made so that the current administrative status better reflects the real work status.

Nominating the INCOSE-AP233 liaison. At the recent INCOSE Symposium, a search was begun to establish a liaison between INCOSE and AP233. The review committee consisted of Jim Schier (TIWG Co-Chair), John Nallon (TIWG Co-Chair), James Martin (Standards Committee Chair), and Dr. Julian Johnson (SEDRES Representative for TIWG), with Mark Sampson, MTTC Chair, as a tiebreaker if needed. A nominee list was developed, and resume(s) and qualifications were reviewed against the agreed-to requirements of the position. The results were reviewed with Dr. John Snoderly (INCOSE Technical Board Chair) for his approval. The AP233 liaison nominating/review committee chose Dr. David Oliver as our liaison to the AP233 effort, and Mr. Doug Stemm as his backup.

These developments call for the formation of an INCOSE group with responsibility for oversight of the liaison activity, and to provide reviews of the AP233 documents and models for completeness and correspondence to the realities of systems engineering. The nominees for the liaison position are particularly invited to participate. The exact relationship between this group and the existing TCs, WGs and technical committee organization will be determined later. Technical committees, working groups and INCOSE members interested should contact David Oliver concerning their interest. It is hoped that this group can become functional at the INCOSE International Workshop in January.

Systems Engineering Applications Technical Committee Activities

William Mackey, wmackey@csc.com

The Systems Engineering Applications Technical Committee (SEATC) is chartered to "Foster the formation and operation of working groups (WGs) and interest groups (IGs) within specific application domains and across domains; and examine systems engineering tools, techniques, and processes within specific application domains." It is the only INCOSE technical committee focused on systems engineering applications in government, academia, industry and non-profit organizations.

The SEATC met this year at the International Workshop in Phoenix, Arizona in January and at the Symposium in Brighton, England in June to review our 1999 activities. We were pleased that all participants worked very well together and accomplished a great amount of work. In Phoenix, all committees were asked to re-examine their goals with respect to commercial and public interest organizations. Also, the SEATC was requested to lead the development of the first 1999 issue of ***INSIGHT***. We readily accepted the opportunity, because we believe that future growth of the discipline depends on commercial and public interest activity.

The article will summarize recent activities of the SEATC. We will continue the excellent work of the following WGs/IGs:

- **Commercial and Public Interest WG (CPIWG)**, led by Mark Austin and William Mackey
- **Facilities Systems Engineering WG (FSEWG)**, led by Pat Sweeney and Ralph Godau
- **Resource Management WG (RMWG)**, led by Ted Dolton and Bill Cutler
- **Telecommunications WG (TELWG)**, led by Tom Bagg and Martin Warner
- **Joint Commercial Aircraft WG (JCAWG)**, led by Greg Mathers and Mary McCartor
- **Environmental Restoration & Waste Management IG** (ER&WMIG), led by Sam Rindskopf and Ralph Hill
- **Railway Transportation IG (RWTIG)**, led by John Williams and Jeff Allan

Two new IGs have been formed as a result of increased interest in these application domains, namely:

- **Motor Vehicles IG (MVIG)** led by Paul Berry
- **Health Care IG (HCIG)** led by John Zaleski

If you believe you have experience or significant interest in one of these groups, please let the chairs know at your earliest convenience. We are also searching for co-chairs for MVIG and HCIG, should anyone be interested.

The SEATC has specific goals for each year, and we work very hard to accomplish them. In order to gauge our progress, the SEATC established criteria in 1997 for the evolution of IG/WGs, and then annually evaluates each group against the criteria. Because the eight-step criteria method may be useful to other INCOSE technical committees, the SEATC is in the process of formalizing this information. As of August 1999, here is how we are doing with regard to our 1998-99 goals.

Goal 1: Improve and modify the Systems Engineering Applications Profiles (SEAP) document for the 1999 symposium and place it on the Web.

Status: The SEAP Version 2.0a, was completed and released in January 1999. The SEAP Version 2.0a is now on the INCOSE Web page under the SEATC products. Also, the University of Maryland has built a Web site as a prototype for the INCOSE SEAP. This effort was lead by Professor Mark Austin, who is currently leading the Commercial and Public Interest WG (go to EE623 under the following URL: http://www.isr.umd.edu/~austin.

Goal 2: Initiate new SEATC work products in all WG/IGs

Status: The following products have been completed by the WG/IGs:

- *SE Applications Profiles Writing Guide*, April 1, 1996
- *Systems Engineering Applications Profiles (SEAP)*, Version 1.0, May 1, 1996
- *Systems Engineering Applications Profiles (SEAP)*, Version 2.0, July 1, 1998
- *Systems Engineering Applications Profiles (SEAP)*, Version 2.0a, January 20, 1999
- List of SE applications papers from previous INCOSE Symposia (Appendices H and I of SEAP)
- Panel session, 1997 symposium, on the topic "Systems Engineering in Commercial Industries"
- Application-focused Symposia sessions for 1996, 1997, and 1998.
- A Facilities SE brochure on the activities of the FSEWG
- Theme issue of ***INSIGHT***, Vol. 1, Issue 2, 1998, entitled "Systems Engineering Application Domains in the Commercial and Public Interest."
- Second theme issue of ***INSIGHT***, Vol. 2 Issue 2, 1999, entitled "Commercial Activities in INCOSE," led by Pat Sweeney
- A Multilevel Participation Plan, 1998, led by Scott Jackson, included as Appendix G of the SEAP
- An Application Domain Template, 1998, developed by Scott Jackson, approved by the SEATC for trial use by the JCAWG to create a Commercial Aviation Guideline Document.
- SEAP Prototype Web pages, 1997, created by the CPIWG, and maintained at University of Maryland
- Regional Seminar titled, "New Arenas for Applying Systems Engineering A Systems Engineering Applications Panel," November 1997, conducted by the RMWG, at a San Francisco

Bay Area chapter meeting.
- Regional Seminar titled "Supersystem Process: Managing Complex Public Issues," January 1998, conducted by the RMWG
- Regional Seminar titled "Systems Engineering in the DOE Environment," October 1998, conducted by the ER&WMIG, in Las Vegas, NV; a second regional seminar was completed during March 1999 in Jackson Hole, Wyoming.

We have done a lot of good work, but we are not stopping. So please plan to help us with items like:
- New SE applications profiles
- Summaries of SE applications papers
- Case studies of SE applications
- List of SE activities and events of other related societies

Goal 3: Conduct Systems Engineering Applications Sessions at the 1999 Symposium

Status: The SEATC Chair worked with Allen Fairbairn, Symposium Technical Chair, to complete this goal. Thirty-five papers were presented at the symposium.

Goal 4: Conduct one or more SE panel sessions at the 1999 Symposium

Status: The SEATC presented three panel sessions at the symposium. They were:

a. "Using Internet for Expanding the Services of Systems Engineering" presented by TELWG
b. "Systems Engineering Aspects of Environmental Restoration and Waste Management" presented by ER&WMIG
c. "A Panel for Railway Case Studies in Europe and the United States" presented by RWTIG

Another panel initially proposed, and now planned for Minneapolis in 2000, is "Issues Related to the Deployment of Systems Engineering in the Commercial and Public Interest Applications."

Goal 5: Continue contact with universities which offer a Systems Engineering curriculum to gain their participation in the SEATC.

Status: Contacts are underway across the nation with systems engineering students and faculty at universities such as Virginia Tech, George Mason University, the University of Maryland, the University of Arizona, and UNLV. Several SEATC members are involved in these activities.

In addition, we began an initiative in 1997 with the University of Maryland, lead by INCOSE member, Prof. Mark Austin to (1) place the Systems Engineering Applications Profiles on the Web, and (2) develop JAVA instructional SE modules for specific application domains. Mark and his students have taken the SEAP concept and created a prototype Web page (see Goal 1). You are encouraged to try the dynamic Java SE Case Studies and perform a modifiable SE tradeoff analysis.

The DOEIG has worked with UNLV in sponsoring at least two regional seminars, with support from the Silver State Chapter in Las Vegas, NV and the Snake River Chapter in Idaho Falls, ID.

Goal 6: Obtain a complement of INCOSE interest groups in local chapters such as:

1. San Francisco Bay Area: Natural Resource Management Systems
2. Washington Metro: Highway Transportation Systems and/or Criminal Justice and Legal Systems
3. Chesapeake: Telecommunications Systems
4. Silver State: Waste Management and Disposal Systems
5. Detroit/Tri-State: Motor Vehicle Systems

Two others are proposed:

6. Texas Gulf Coast: Energy Systems
7. New England: Health Care Systems

Status: The first five chapters are conducting or proposing programs in their local chapters. The San Francisco Bay Area Chapter has had as many as eight volunteer projects underway in Natural Resource Management. In October 1998, the Silver State Chapter conducted a Workshop entitled "Systems Engineering Within the DOE Complex."

Goal 7: Improve Team Building and Communications in the all SEATC WG/IGs.

Status: It is apparent that lack of resources, limited commitment in a volunteer organization and downsizing in many industries have caused a few people to carry the burden on virtually all INCOSE committees. The SEATC is fortunate in that its people have continuously demonstrated that they are willing to work well together and to produce materials useful to the SEATC and to INCOSE. Nevertheless, we in the SEATC believe we can continue encouraging new participation in all WG/IG activities.

Conclusion: SEATC conducts regular telecons with all WG/IG chairs. Using a structured agenda, these calls last one hour and to date all business has been conducted. The time of the calls is 11:00 am U.S. Eastern Time, which permits participation from Europe to the U.S. West Coast. In addition, the JCAWG currently conducts telecons almost weekly in order to develop the new Commercial Aircraft Guideline.

If you like exciting activity and are interested in SE applications, please join one of our WGs/IGs. Please contact

- William Mackey at wmackey@csc.com, 301-794-2138
- Scott Jackson at scott.jackson@boeing.com, 562-496-5049
- Ralph Godau at rigodau@rmit.edu.au, 0412-294-541

I wish to thank all of the SEATC members who contributed to the realization of all our 1999 goals, and those who stimulated us in Phoenix and Brighton to continue the good work in 2000. I hope to see many of you in Scottsdale, AZ for the International Workshop.

News from Chapters

Summer School on Systems Engineering and Supportability Analysis, held in Utö, Stockholm Archipelago

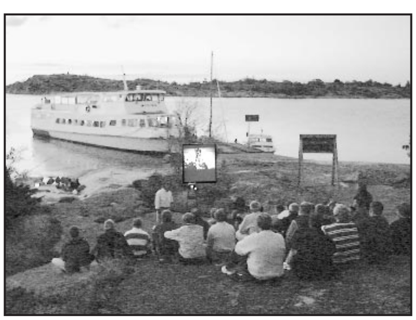
Tom Strandberg, strandberg@syntell.se

Twenty-five delegates from Sweden, Norway and the U.S. that participated in the first Scandinavian Summer School held August 15-20 on Utö, one of the largest islands in the Stockholm archipelago. The delegates representing sectors such as automotive, aerospace, defense and transport were challenged by a program with sessions stretching from 8:30 AM till 9:00 PM. Enthusiastic lecturers, including Dr. Dinesh Verma (Lockheed Martin) and Dr. Jedzimir Knezevic (MIRCE Akademy), conducted the classes. It was a hectic week, but the reward was large.

To facilitate the assimilation of the information, we also took part in activities that stimulated the other half of the brain. We will always remember sitting on the rocks of a small island watching the sun set and listening to a presentation about "stepping out of our zone of comfort," whether it is climbing the Mount Everest, breaking the record sailing across the Atlantic, or promoting SE and supportability issues within our organizations.

The class's evaluation of this first-time event gave the organizers a firm basis for conducting a second Summer School on SE. Next year's event will be held in Norway with the support of Kongsberg. Those who are interested in attending this intense training and become part of a growing network of practitioners are invited to contact Tom Strandberg, Syntell AB, strandberg@syntell.se, +46-8-660 02 80.

Summer School Class of 1999



Thinking outside the boundaries.

The Brainwash

Midwest Gateway

John Hulsman, Jr., Secretary
john.r.hulsman-jr@boeing.com

The Midwest Gateway Chapter's June event featured Mr. John Vu, who spoke on software process improvement initiatives. This event brought home the necessity of a systems approach to complex development projects. Mr. Vu's presentation made the point that to get to the higher maturity levels for software development processes, you must take into account more than just software processes; i.e., you can't improve software processes without improving the processes involved with all of the systems that interact with that software. Focusing on just one piece of a system won't get you a SYSTEM that works.

The July event, held at the Engineers' Club of St. Louis, featured Bill Schoening speaking about significant events at the recent International Symposium in Brighton, England. Bill discussed some of the new initiatives in the technical and administrative aspects of INCOSE. He also discussed the systems engineering approaches used by several European companies, encompassing both the military and commercial markets, which he has visited over the past several months.

On September 18, the chapter sponsored a tutorial titled "Object-Oriented Systems Engineering" at Washington University. David Beshore of Boeing Rocketdyne, and Vice President of the INCOSE Los Angeles Chapter, made the presentation. The tutorial taught fundamental object oriented techniques using class modeling, use cases, and scenarios to better define systems and system models. At the conclusion of the tutorial, attendees were able to more accurately define customer needs and model a system, and better explain their efforts to other engineers and management.

San Francisco Bay Area

Dorothy McKinney, President,
dorothy.mckinney@lmco.com

Beginning in January, our monthly chapter meetings have been held at Lockheed Martin Missiles & Space in Sunnyvale. The presentations from the middle of 1999 were the following:

- April – Improving Systems Engineering Career Prospects In A “Better, Faster, Cheaper” World, by John Hoschette, Lockheed Martin
- May – Best Practices Guide and Case Study Examples from the French Space Agency, by James H. Brill
- June – Potomac Fever or Potomac Fog, Or What Is a Systems Engineer Doing on Capitol Hill?, by Frederick Martin
- July – The Yucca Mountain Project, Finding a Suitable Site for Spent Nuclear Fuel and High-level Radioactive Waste, by John Clouet, TRW
- August — Designing Lighter Methodologies, by Alistair Cockburn

Our chapter’s “core series” of tutorials continued:

- Decision Making and Risk Management — Key to Implementing Systems Engineering by Barney Morais & Dr. Brian Mar in April,
- Secrets of High Performance Project Teams — Tools for Building and Maintaining High Technology Teams by Michele Jackman in May, and then a summer vacation.

The tutorials have resumed in September. Sponsoring a comprehensive core series is an ambitious undertaking for the chapter and is made possible due in large part to the leadership of Mr. Bob Barter, Chapter President-Elect, and the chapter board. Please visit our Web site for the latest news on our tutorials.

Speaking of the chapter Web site, it can be found at its permanent location on the INCOSE server at http://www.incose.org/sfbac. Please visit our members area to see what makes our chapter special to its membership.

Over the years, we have found that offering our members an opportunity to work on a joint activity advances the opportunities to network as well as professional development. The *Systems Engineering Handbook* was the chapter’s first group project. At the 1999 International Workshop in January, we received concurrence from the Technical Board to lead the effort in revising the handbook. Jim Whalen <jtwhalen@earthlink.net> and I will take the lead as editors of the handbook, with Dick Wray coordinating inputs from other chapters. We solicit the assistance of INCOSE members in producing the new edition. Please contact Dick Wray or the lead editors to volunteer.

Bob Barter has begun working with other local systems engineering and program management professionals, as well as local universities, to determine if there is enough demand in the San Francisco Bay Area to support a local systems engineering certificate program. The assistance of other chapters who have already started such efforts has been very valuable.

Our upcoming events are:

- September 25: Tutorial, Systems Architecting, by Dr. Mark Maier
- October 12: Monthly meeting
- October 23: Tutorial: Engineering of Complex Systems, by Dr. Brian Mar and Barney Morais
- November 9: Monthly meeting: The Web Based Masters Program in Systems Engineering by Herm Migliore, Portland State University
- December 14: Monthly meeting (tentative), A System Engineering Looks at The POTS (Plain Old Telephone System), by Tom Jackson

All SF Bay Area Chapter meetings are held at Lockheed Martin Missiles & Space in Sunnyvale at 5:30 p.m. Check our Web site for announcements and directions: http://www.incose.org/sfbac. All SF Bay Area Chapter sponsored tutorials are held at San Jose State University on Saturdays, 8 a.m. to 5 p.m. Pre-registration is required.

Call for Papers

Call for Papers

Systems Engineering: People, Processes, Technology, and Systems

International Council On Systems Engineering (INCOSE) Mid-Atlantic Regional Conference

April 6-8, 2000 • Sheraton Hotel – Reston, Virginia

Sponsored by the Washington Metropolitan Area, Central Virginia, Chesapeake, Hampton Roads, Liberty, and Southern Maryland Chapters of INCOSE

Original papers are requested on topics related to the SE: People, Processes, Technology, and Systems theme. Submittals from industry, government, and academia are solicited. Submittals from students are encouraged as we are planning for student paper sessions. Some candidate program topics include:

- Systems Engineering Processes
- Information Technology
- Person and System Interfaces
- Applying SE in Customer Services
- Making Processes People Friendly
- Using SE in Web Site Design
- Use of Internet Technology
- Systems Engineering Training, Mentoring, or Education
- Business Process Engineering/Reengineering
- System or Process Integration Issues and Challenges
- Use of Tools, Modeling, or Simulation to Facilitate Integration
- Practical and Theoretical Approaches for Managing Integration
- Using SE in the Management of Data and Information
- Case Studies and Lessons Learned

Submittal Requirements For Paper Summaries:

Submit a 2-4-page paper summary. Include: 1) title, author(s) and affiliation(s), and brief biographical sketch; 2) a brief abstract (~50 words); 3) a concise description of the approaches or methods used – emphasizing elements that are important, unique or innovative; 4) a summary of the main points, conclusions drawn, and/or lessons learned; and 5) contact information for the primary author – including name, affiliation, address, email, and phone number.

It is our intent to work and communicate primarily via email. Submittal as an MS Word document is strongly preferred. Address paper-related questions to the Technical Program Chair.

Send Paper Summaries to: Pohlmann-wma@erols.com

Schedule:

• Call for Papers Issued	May 20, 1999	Final Papers Due	Jan. 31, 2000
• Paper Summaries Due	Sept. 15, 1999	Presentation Materials Due – Hardcopy Plus Electronic	
• Notification of Acceptance	Dec. 1, 1999		March 15, 2000

Technical Program | **Chair Conference Chairs**

Dr. Lawrence D. Pohlmann
Strategics Consulting
(703) 406 2595
Pohlmann-wma@erols.com

Ms. Dona Lee
Dynamic Systems
(703) 684-4060
Donalee@dynsys.com

Mr. David Long
Vitech Corporation
(703) 883-2270
Dlong@vtcorp.com

Additional Information on the Conference Web Site: www.incose-marc.org

INCOSE

Workshop

Central Florida

SYSTEMS & SOFTWARE ENGINEERING: FROM THEORY TO PRACTICE

A One Day Tutorial

November 6, 1999

Instructor: Dorothy McKinney

This tutorial is intended for systems engineers, software engineers and engineers in other disciplines who work with software engineers who need to understand the process by which software engineers get their requirements and guidance to make key software architectural decisions and design trades.

No software implementation knowledge or experience is required for the course. What is required is the willingness to take risks and experiment with the process of developing requirements and derived requirements during the tutorial.

The seminar modules will be as follows:

Module 1: Understanding the Big Picture
Module 2: Using Existing Software, Including Commercial Off the Shelf (COTS) Software
Module 3: Theory and Practical Application: Requirements Management, Baselines and Configuration Control, Metrics and Tools
Module 4: Defining Detailed Requirements for Software
Module 5: Costs of Change
Module 6: Using New Techniques: Rapid Prototyping and Object Oriented Development

- Date: Saturday, 6 November 1999
- Time: Registration starts at 8:00 a.m., workshop at 9:00 a.m.
- Place: USF St. Petersburg, Bayboro, Campus
- Cost: Space is limited. Be sure to enroll early. Early registration deadline is October 1, 1999.

	Early Registration	Late Registration	INCOSE membership
INCOSE Member:	$50	$60	NA
Non-Member:	$60	$75	+$30
Student (full time)	$25	$40	+$5

(* The INCOSE Membership is good through May 2000. Annual membership is $80.)

For additional Tutorial and Registration information, contact Ben Berauer, 727-302-7693, bfbc@eci.esys.com. More complete information, including an application, can be received by email, or at our web site, www.netcom.com/~rlmrchnt.

I2K Bug
Avoid It Now !!

As we near the year 2000, we need to make sure that we are not only Y2K (Year 2000) compliant, but also I2K compliant. To check your compliance, please go through the following steps:

1 Turn your calendar or date planner to the year 2000
2 Look at the month of July
3 Check the activities you have planned for the days 16-20
4 If you do not have the INCOSE 2000 International Symposium marked on your calendar, then you have the I2K (INCOSE 2K) Bug.

Repair:

1 Mark your calendar for the INCOSE 2000 International Symposium on July 16-20, 2000. This will take place in Minneapolis, Minnesota, USA.
2 Write "Make INCOSE 2000 Symposium Reservations" on your calendar some time before July.

Congratulations! You are now I2K compliant.

To learn more about the INCOSE 2000 International Symposium, our web site is alive and active. Please take a few minutes and check us out through the main INCOSE web site at http://www.incose.org, or at: http://www.incose.org/nrthstar/i2k/index.htm. This web site will be your quick access point for schedule information and can be used as a means for registration. The site contains calls for papers, exhibitors and tutorials, as well as information about travel and hotels in Minneapolis. Please note that draft papers must be submitted by November 1, 1999.

For those who stopped by our booth at the INCOSE 1999 International Symposium in England, we hope you found Minneapolis an attractive place to visit. The symposium plans to be rich in technical content and presentation.

Please consider attending the symposium, submitting a paper, and becoming an exhibitor for INCOSE 2000. You *will* have an impact on the "New Century of Opportunity."

As for the I2K Bug – Avoid it now!

Conference on Systems Approach to Product Innovation and Development in Hyper-Competitive Environments

(INCOSE Colorado 2000)

Hilton Denver Tech South • Denver, Colorado USA • March 26-28, 2000

This event is sponsored by the Colorado Front Range Chapter of INCOSE. The focus of the conference is to provide a forum for diversity of thinking as promulgated by the INCOSE Strategic Directions document dated June 1999. The goal is to explore systems engineering (SE) best practices in distinctive competitive product innovation and development within the context of a more diverse systems practitioner community. This conference welcomes dialogue among the traditional SE aerospace and defense companies and other industry organizations working on complex issues of managing, creating, and launching products and services to the hyper-competitive global marketplace.

Scope of Conference: The two and one half-day conference will include tutorials, technical presentations and networking within the exhibition area. English will be the working language. Emphasis will be:

- Tools & Methodology
- Product Development Lifecycle
- Reduced Time to Market
- Enhance Product Customization, Quality and Market Acceptance
- Product Development Metrics
- Product Portfolio Management
- New Product Forecasting

Attendee Profile: Product Developers, Technology Transfer Agents, Project Managers, Systems Integrators, Software and Hardware Engineers, Continuing Technical Educators and Professionals, System and Database Architects, Systems Engineers

Call For Papers: Authors wishing to contribute to the conference should send an abstract (approximately 500 words) of their proposed contribution. Electronic submission of abstracts is preferred. Submission should be in Microsoft Word, WordPerfect or Text. Your abstracts (due by November 15, 1999) should be compiled in the following format:

- Title of Paper, Use correct punctuation at the end.
- Author's full name using first name first. Use a comma after the full name.
- Business name. Use a comma after the business name.
- Your business mailing address (address, city, state or province, zip, country).
- If more than one author, separate by semi-colon, after each author's mailing address. Underline the speaker's name if more than one author.
- SPACE
- Abstract of approximately 500 words
- SPACE
- Indicate name, business, mailing address, phone, fax and e-mail for the primary contact.

Call For Tutorial Proposals: The Program Committee is also soliciting proposals for half-day tutorials before October 1, 1999. Electronic submission of Tutorial Proposals should be sent to milestone@bod.net. Proposals should contain a complete list of the instructors, their addresses, titles and one page abstracts of the tutorial objectives. The organizer should also attach a cover letter describing the main theme of the proposed tutorial, planned instructional methodology and delivery format.

Call for Invited Sessions: The Program Committee is also soliciting proposals for invited sessions and papers before November 1, 1999. Electronic submission of Invited Sessions and papers should be sent to milestone@bod.net. Proposals should contain a complete list of the authors, their addresses, titles and one page abstracts of the paper and session title. The organizer should also attach a cover letter describing the main theme of the proposed session.

Deadlines:

- Submission of Abstracts: November 15, 1999
- Submission of Tutorial Proposals: November 15, 1999
- Submission of Invited Sessions: November 15, 1999

Conference Committee:

- D. Alex Chuang (Conference Chair), ICG Communications, Inc.
- Prof. Don Clausing (Program Chair), MIT, Center for Innovation in Product Development
- Leonard "Lenny" E. Mell (Conference Co-Chair), Pathfinder Solutions, LLC
- Other Members: To Be Announced

Abstract and Proposal Submission:

INCOSE Colorado 2000
C/o Milestone Presentations, LLC
4255 S. Buckley Road, Suite 118
Aurora, CO 80013 USA

Phone: (303) 690-3233
Toll Free: (800) 996-3233
Fax: (303) 690-3278

E-mail: milestone@bod.net
Web: http://www.milestoneshows.com/incose/
Contact: Mark Stone

INCOSE Infrastructure

Volunteering As You Like It

Membership Committee Co-Chairs: Lew Lee (lew.lee@trw.com) and Dona Lee (donalee@dynsys.com)

INCOSE has many volunteer opportunities around the world. There are many positions already available, as well as positions that you can help create. Take the time to consider the possibilities. Here are some questions to help you identify your interests:

■ **What issues matter the most to me?**
Is there a Technical Committee or chapter program formed to address your issue? Is the committee active with clear goals? Is your area of interest underrepresented or not well addressed? Could you lead a new effort? These questions will help you to focus on committees of potential interest. Take a look at what the Technical Community is doing: http://www.incose.org/techcomm.html

■ **Shall I volunteer for something that uses my existing skill set? Maybe it's time to try something new?**
What are you good at doing? Look beyond your career skills. What are your hobbies and extracurricular interests? Be sure to consider the following: Do I want to use these talents in a volunteer capacity? Would it be more of what I do at work? Are these skills I'm interested in improving? You may want to volunteer for something that offers a "change of pace" from your daily routine. Visit the Chapters page to find out more about activities on the local level: http://www.incose.org/chap.html.

■ **Is there an opportunity for me to lead?**
Perhaps you'll have the chance to experiment with new techniques. Perhaps you'll expand your management skills set. Look at this as a way to have fun or an opportunity for a fresh perspective. Opportunities abound to "learn while leading." Find out more about the responsibilities of the Officers and Board of Directors: http://www.incose.org/who.html

■ **How much time can I contribute?**
Shall I seek a one-time assignment? How about a short-term assignment, or an ongoing assignment? Studies of those who volunteer have identified a clear trend: "I want to volunteer but I can do it only if it's on my schedule." Perhaps one evening a month; maybe one hour a week. Be honest with yourself and set realistic availability goals. This way the organization can count on you and you'll be satisfied with your accomplishments. It's much easier to expand volunteer hours than cut back.

■ **What's on my "not interested" list?**
Admit it, there are things that are on your No-No list. Get these identified and use them to help you find an agreeable position.

■ **With what kind of people will I be working? Who will benefit from my efforts? Who might be on my team?**
INCOSE members tend to be well-versed in both work-related and non-work areas. Opportunities abound to be a mentor and to be mentored. Collaborate on a Technical Committee work product. Build your network of professional contacts.

■ **Do I want to work alone or with a group?**
There are large committees and there are small. There are tasks that are better done by one than by committee. At the local level, there are lots of small jobs which require a conscientious volunteer, such as: posting meeting announcements on corporate electronic bulletin boards, mailing flyers, or helping at the monthly program registration desk. Work with your chapter to make it a stronger organization.

Seeking a volunteer position is very much like looking for a job—with a significant upside! You can expect volunteering to be an enjoyable way to spend your valuable time, with the added benefits of increasing your professional standing in the marketplace, expanding your network of contacts, and helping INCOSE. So take the time to ask yourself a few questions and focus on how to get yourself involved.

Bob Kenley and Terry Creque, New Ways & Means Co-Chairs

Ken Ptack, ptack_ken@prc.com

I am pleased to announce that Bob Kenley accepted the position of Ways & Means Chair in earlier this year. Recently accepting as co-chair is Terry Creque, who will hold the position for two years. Bob and Terry will be working closely to maintain procedural structure and order for organization and operation in accordance with INCOSE Policy WMC-100. Bob can be reached at bkenleyx2@aol.com and Terry can be reached at creqtr@inel.gov.

Call for Nominations for INCOSE Fellows

Terry Bahill, terry@sie.arizona.edu
Chair, INCOSE Fellows Selection Committee

The Fellows Selection Committee will be pleased to accept nominations for new INCOSE fellows. Nominations may be made by INCOSE members or by INCOSE fellows.

Nomination packages will be accepted until 1 December 1999. Final discussions by the INCOSE Fellows Selection Committee will be held at the INCOSE International Workshop in January 2000. This committee will submit a list of recommended fellows to the INCOSE Board for the April Board meeting. New fellows will be announced at the International Symposium in Minneapolis, Minnesota, USA, 16-20 July 2000.

The INCOSE Fellows Selection Committee is composed of: Elliot Axelband, A. Terry Bahill, Ben Blanchard, Wolt Fabrycky, George Friedman, James Martin, Andy Sage, Richard Stevens, John Velman and A. Wayne Wymore. Please submit Fellows nomination packages and requests for the Letter of Support form to:

Terry Bahill
Chair, INCOSE Fellows Selection Committee
Systems and Industrial Engineering
University of Arizona
PO Box 210020
Tucson, AZ 85721-0020
terry@sie.arizona.edu

The following is the official INCOSE Fellows Award Policy.

1. Fellows are a special class of membership within INCOSE. Selection of Fellows shall be by the Board of Directors upon recommendation of the Fellows Selection Committee. Membership in the Fellows class will not exceed one percent of the total membership. About six new Fellows will be selected each year until the maximum is reached. After that the maximum number selected each year will be approximately 0.1% of the total INCOSE membership.

2. Fellow Award Eligibility. Candidates must have been INCOSE members for a minimum of five years. Under exceptional circumstances, this can be waived by the Board of Directors.

3. Fellow Award Criteria. Fellow awards are based only upon significant verifiable contributions to the art and practice of Systems Engineering, and only upon evidence of same, provided by their nominators in written form to the Fellows Selection Committee.

   It is recognized that systems engineers come from different domains, e.g.: industry, government and educational organizations. They also are engaged in different areas of practice, including, research, application and teaching. In some cases, national security or company policy inhibits accessibility of supporting materials. Therefore, varied verifiable evidence of contributions to the state of the art and practice are expected to be submitted.

   Nominators should identify their candidate's primary strength as that of either a practitioner (applies knowledge), or a researcher (develops new knowledge), or a teacher (imparts knowledge to others). For a practitioner, the criteria are satisfied by providing evidence about programs that he/she has personally led and/or advanced by means of significant application of the systems engineering art. This evidence should be supported by publications — ideally in refereed journals or conferences where possible — or other suitable means.

   For a researcher, the criteria are satisfied by providing evidence about research personally conducted or advanced as a consequence of the researcher's effort. This evidence should be supported by patents, patent applications, books authored and those to which contributions have been made, and publications in refereed journals or conferences. For teachers, evidence is provided by advances made in the state of the art in systems engineering education such as new books, courses, curricula and refereed publications.

   Some nominators may wish to submit their candidates for consideration in more than one category. In this case, evidence must be provided as above for every applicable area.

4. Fellow Award Process. Each candidate will have a nominator other than him- or herself. The nominator will provide a package to The Fellows Selection Committee that will consist of the following:

   A) *Candidate Profile:*
   Name of Candidate
   Year of Birth
   Primary Contribution
   Secondary Contributions (if applicable)
   Citation (to be used on certificate)
   Educational Background
   Professional History (Employer, Years of Employment, Duties, Accomplishments)
   Accomplishments vs. Fellows Criteria

   B) *Letters of Support.* Letters of Support must be provided by at least three people recommended by the nominator. All letters must be from active members of INCOSE, and at least one letter should be, if possible, from one of the current INCOSE Fellows. Those writing letters of support should have the candidate's resume available to them, but each letter of support should be independently written. Letters of Support should use the standard form, which is available at http://www.sie.arizona.edu/sysengr/ INCOSE, at the INCOSE web site and on the next page. Letters of Support should be mailed directly to the Chair of the INCOSE Fellows Selection Committee, and must be received on or before December 1, 1999.

   C) *List of Supporters.* The nominator will contact supporters directly to have them write letters of support. The nominator will submit a list of up to five people who have been asked to submit letters of support for the nominee. This list should be provided in the nomination package.

HANDWRITTEN COPY IS NOT PERMITTED

INCOSE 1999 LETTER OF SUPPORT FORM

NAME OF CANDIDATE ______________________________
LAST, First, Middle

NAME OF SUPPORTER ______________________________
LAST, First, Middle

NAME OF NOMINATOR ______________________________
LAST, First, Middle

A. If you are not qualified to judge the work of the candidate, please check this box ☐ and notify the nominator immediately.

B. How long have you known the candidate and in what capacity? ______________________________

C. On the basis of the work of the candidate, which you are competent to judge, please indicate whether or not, in your own judgment, the candidate meets the requirements for Fellow grade. What distinguishes this contribution from the norm?

D. **CHECK AT LEAST ONE BOX**, identifying the area of the most significant contribution which qualifies the candidate for Fellow grade:

☐ Practitioner ☐ Researcher ☐ Teacher

E. INDICATE BY AN "X" below where the individual contributions of the nominee fall in the qualifications for Fellow grade.

Not Yet Qualified	Marginally Qualified	Qualified	Highly Qualified	Extraordinarily Qualified
0	3	5	7	10

F. ______________________________
Date Your Signature INCOSE Membership Number

Business Affiliation

Street Address E-mail Address

City /State /Province Zip /Postal Code Country Tel. No. (Incl. area code) Fax No. (Incl. area code)

G. Are you a Fellow of INCOSE or any other professional society? If so, which society? ______________________________

H. Please include a brief resume of your career.

I. Please return this form to:
Terry Bahill
Department of Systems and Industrial Engineering
The University of Arizona
1127 East North Campus Drive
Tucson, AZ 85721-0020

Commentary

Advances In Commercial Product Development Lessons for INCOSE Systems Engineering

Don Clausing, Elliot Axelband, and R.B. Campbell

Introduction. The authors believe that commercial product development methods can improve the current practice of (defense/aerospace) systems engineering and – with the encouragement of many INCOSE members – have written this article to stimulate relevant information and practice exchange.

Systems engineering as a formal discipline began in the Bell Labs during the 1930s. Its practice received a great boost during World War II. INCOSE is dominated by defense/aerospace related engineers. As a result, INCOSE-legacy systems engineering is rooted in the defense/aerospace industries and other industries in which competition is constrained by external rigidities and a limited number of competitors.

Meanwhile in the heartland of the commercial industries in the United States, the end of World War II brought an unnatural competitive situation. Pent-up demand, after 16 years of economic depression and war, and the destruction of other major economies, created an abnormal supply-push economy in the United States.

In these circumstances both INCOSE-legacy systems engineering and commercial new product development institutionalized processes and practices that seemed excellent unto themselves, but which contained serious weaknesses. Each practice tended to be an elegant solution to a received question. However, in retrospect the questions that were asked were somewhat under conceptualized.

To cite an example from the production world, the materials-management specialists worked on the traditional problem – better storage and retrieval of inventory. Their elegant solution was high-bay, automated storage-and-retrieval buildings. Then the United States learned about the Toyota Production System – lean production. As a Hewlett-Packard pioneer in 1983 said, “All we could do with those elegant buildings was take a bull-dozer to them.”

In summary, in the late 1970s INCOSE-legacy systems engineering and commercial product development had many strengths. However, the quality, cost, and schedule usually left significant opportunities for improvement.

New Competition in the Commercial World.

In the heartland of the commercial world, the previously unnatural situation in the United States changed rapidly during the 1970s. This was largely the result of new competition that came from the best companies in Japan. The response to the new competition from Japan varied greatly. Some companies ran to Washington for protection. Some tried to shift into less competitive industries. Some companies, however, went to study the best Japanese companies, and launched major improvements. As a result of this intense global competition during the past two decades, commercial companies have made great improvements in product development, many of which are essentially improvements in systems engineering. INCOSE-legacy systems engineers and companies could benefit greatly from these improvements.

Lessons for Systems Engineering.

In commercial new product development, some of the most important improvements that have been implemented are in the domain of systems engineering. Importantly, in the commercial world these have not been referred to as improvements in systems engineering, but rather as improvements in new product development.

In 1988, the MIT Commission on Industrial Productivity (including one of the authors – Clausing) completed its two-year study, which was published as the book *Made in America, Regaining the Productive Edge*. The MIT Commission found six weaknesses in manufacturing industries in the United States. One of these was Technological Weakness in Development and Production. This weakness was rooted in underlying problems in systems engineering/product development. The Commission noted the improvements in problem prevention and customer focus that were being implemented by leading commercial companies in the United States.

Also in 1988 the Under Secretary of Defense for Acquisition recognized that the Department of Defense could benefit from the improvements in product development that were being implemented by leading commercial companies. Therefore, a major study was commissioned that was carried out by the Institute for Defense Analysis (IDA).[1] Subsequently he formed the Defense Manufacturing Board, which then formed the Concurrent Engineering Task Force (CETF).[2]

The outcome of this 1988–1991 activity has been the introduction of Integrated Product Development (IPD) into the weapons-systems industries, and thus into INCOSE-legacy systems engineering. The IPD that is now practiced as part of INCOSE-legacy systems engineering is basic IPD, often a simple form of basic IPD. The total-quality practices that build on IPD to constitute the most advanced practice of new product development in the commercial heartland have still not been widely integrated into INCOSE-legacy systems engineering.

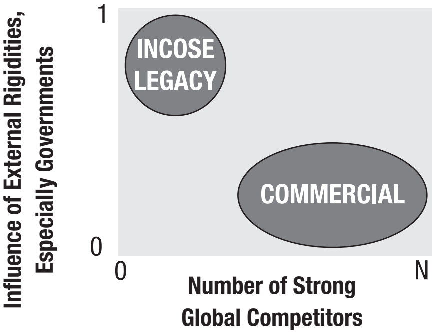


Figure 1. Distinction between INCOSE-legacy companies and commercial companies.

INCOSE Legacy versus Commercial

The fundamental distinction is sometimes referred to as defense versus commercial, or public versus private. However, neither of these comparisons adequately captures the issue. The two essential differences are displayed in Figure 1 (above).

The INCOSE-legacy companies are strongly constrained by external rigidities, often the government. Usually there are few, if any, global competitors. The commercial companies are at the opposite end of both spectrums: few external rigidities and significant global competitors.

Many companies that are INCOSE-legacy companies are nominally commercial companies, and some even have the word commercial in their name. Nevertheless, they are positioned in the upper left corner of Figure 1, and therefore are not commercial as we are using the term in this article. In this article a commercial company is defined as one that is in the lower right corner of Figure 1, that is a company that faces sufficient strong global competitors and is little constrained by external rigidities. Xerox is an example of the distinction in Figure 1. In the 1960s Xerox was in the upper left corner. The external rigidity was a patent monopoly. As a result there were not any competitors. A patent monopoly does encourage innovation, but it also subsequently encourages weak practices. Xerox was attacked by vigorous global competition starting in 1975. During the 1980s and the 1990s Xerox made great improvements in its practice of product development, and regained market share. By implementing these improvements Xerox has moved from the upper left of Figure 1 to the lower right. Xerox is now a leader in product development practice.

The Situation Today

Today INCOSE-legacy systems engineers walk into the forest of the commercial heartland and ask, “Where is your systems engineering?” They would be better rewarded if they asked to benchmark the product development processes and practices at companies such as Xerox and Hewlett-Packard.

Commercial Viewpoint

Meanwhile the product development leaders in the commercial heartland look at INCOSE-legacy systems engineering, and their reaction is that this is where they were back in the 1970s before they understood their problems. The major weaknesses in INCOSE-legacy systems engineering as viewed from the commercial heartland are:

1. Too much dependence on problem reaction, rather than problem prevention
2. Too much internal focus; too little attention to the customers, especially the ultimate users
3. Too many (arbitrary) constraints
4. Too little emphasis on cost, schedule, and productivity

Systems Engineering Outreach

The INCOSE-legacy systems engineers are reaching out to the commercial world in a limited way. This primarily consists of:

1. Elimination of bureaucratic reporting requirements that had been imbedded in government contracts
2. Reaching out to those systems engineers in commercial com-

panies who have an affinity for INCOSE-legacy systems engineering

Both of these are very welcome. However, there is still the opportunity to reach out to the commercial heartland, where the best product development is being practiced.

INCOSE Posture

INCOSE seeks to facilitate the adoption of relevant systems engineering practices in defense systems engineering and to enlarge its constituency to include both defense systems engineers – by far the current majority – and commercial new product developers within its ranks. But alas, INCOSE looks inward, and reinforces the INCOSE viewpoint. Nearly all of the directors are from the INCOSE-legacy world. Most INCOSE members are not familiar with the commercial world. The current situation is summarized by two comments that were heard at the INCOSE 1997 Symposium:

1. "INCOSE views the commercial world as being inferior – INCOSE will teach them how to do it right. Instead INCOSE should approach the best of the commercial world with an attitude of learning."
2. "It is a myth that commercial people do it better than DoD companies."

Comparison: INCOSE-Legacy versus Commercial

The following differences are fundamental and generic (shown below).

In addition to the generic differences, there are further strong differences when the rigidity is a result of the customer being the government.

INCOSE LEGACY WORLD	COMMERCIAL WORLD
"Best practices" aim at satisfying external rigidities	Competition improves best practices
Good answers to wrong questions	Large shift in last 20 years from problem reaction to problem prevention
Small improvements during the last 20 years	Performance metrics have improved greatly during the last 20 years
Contract compliance is king, time cycles long	TTM (time to market) is king
Competition constrained to bidders	Competition everywhere
Bureaucratic, massive infrastructure, complex rules	Simple infrastructure

	GOVERNMENT	COMMERCIAL
Market	Market defined by procuring agency. Procuring agency segments market. Procurement quantity sets volume. Relatively small marketing/sales staff.	Market defined by competitive pressures. Competitors can segment and even shape markets. Market share central competitive measure. Relatively large marketing/sales staff.
Customer	Single (monopsonistic) customer, albeit multi-faceted. Customer defines his requirements. Customer frequently provides service and support	Customers are varied and frequently large in number. Customer "requirements " have to be established. Developer responsible for sales, services, and support
Competition	Intense until award; minimal thereafter. Frequently national champions only. Limited set.	Continuous and unending. Frequently worldwide. Often large in number.
Product Development Process	Structured by bid process. Contractual relationship. Hard and complete specifications at start. Formal change process. Many key top-level parameters fixed.	More degrees of freedom in process structure. Largely informal internal relationships. Specifications can be developed on a just-in-time basis. Organizational empowerment for many changes. Most parameters uncertain or can be traded off.
Scale/scope	Large-scale, high technology projects common. State-of-the-art technologies. Success criteria primarily non-financial.	Scale and scope constrained by competition and budgets. Technology readiness a constraining factor. Commercial return on investment mandatory.
Risk	Primarily technical. Technical performance dominating requirements.	Primarily market. Complex technical, cost, and quality trade-offs.

Source of Problems

How do rigidities and lack of competition cause problems? In three fundamental ways:

1. Encourages resistance to change
2. Excellent engineering, but …
3. Lack of competition reduces the incentives

We all have a natural resistance to change. "It worked there, but it won't work here, we're different." The cloistered INCOSE-legacy environment encourages this natural tendency.

Excellent engineering is strongly defended. It is excellent engineering; it provides excellent answers to the INCOSE-legacy questions. What is ever so difficult to realize for the engineers who are working in the upper left corner of Figure 1 is that the questions are wrong. A production example has already been mentioned. "How do you best store and retrieve inventory?" It sounds reasonable, and many organizations had provided elegant answers to it. However, it was the wrong question.

Wrong questions abound in the INCOSE-legacy systems engineering world also. "How do we best predict field reliability during the development phases?" Again it sounds reasonable, and again it is the wrong question. The whole traditional field of reliability has grown up around this and related questions. The right question is, "How do we most rapidly improve reliability during development?" It is much better to improve reliability by 200% with fuzzy precision, rather than have precise knowledge that reliability has been improved by 20%.

To dig into the issue a little more, we can achieve precise knowledge of reliability by holding the design configuration constant, and running many repetitions of tests to reduce the confidence interval. That will not improve the reliability, because the design is not being changed. Or alternatively, we can keep changing the design in a systematic way to improve the reliability. Because we do not run many repetitions for any one configuration, we do not have precise knowledge of the reliability of any configuration. However, the reliability will be greatly improved. This is one example among many of excellent engineering in the INCOSE-legacy world – to the wrong question.

The lack of competition fails to provide feedback signals that the practices need to be improved. Engineers at Xerox in the 1970s thought that they were doing a great job. Engineers in Japan were able to achieve comparable performance for far lower cost and in much shorter time. By 1981, competitive inroads in market share had led Xerox engineers to the realization that huge improvements were needed.

Commercial Product Development

The revolution in new product development that has occurred in the commercial world during the last two decades features concurrent engineering, QFD with focus on the customer, robust design, Pugh concept selection, reusability with portfolio planning, technology-development readiness and management, design for lean manufacturing, simultaneous optimization of cost, performance, and schedule, TRIZ, and effective phase gates with meaningful criteria. These now provide quality, cost, and schedule that are far superior to those that were standard 20 years ago in the commercial world.

Can Defense Systems Engineering Change?

Defense systems engineering will change but only when it realizes it is in its best interests to do so, and the government truly gives it the freedom to act in this way. While defense systems engineers, in many cases, would like to introduce change, and have in fact brought about many positive changes to date, the extent to which change is possible is limited by the regulatory policies of the DoD.

In the 90's, the DoD launched the Acquisition Reform Movement. Its intent was to adopt commercial practices by minimizing government oversight and streamlining the acquisition process, eliminating unnecessary requirements. The government was no longer going to specify how to build its products, merely what it wanted built, allowing new freedom for industry. Alas, this posited "Revolution in Business Affairs," while having made some progress, has fallen far short of its aims. In fact in the summer of '99 one senior DoD official toured Europe encouraging foreign acquisition of U.S. defense firms – hardly *laissez faire*.

Two Ships in the Night

Today, we have two ships that have passed in the night. INCOSE-legacy systems engineering has sailed out from the monopsony port looking for systems engineering. The commercial heartland has sailed from the port of the international competitive wars, constantly improving product development. The two ships have passed with little recognition. A synthesis that integrates the best of both worlds seems to be in the best interests of everyone.

What Should INCOSE Do?

The authors, two of whom are INCOSE Directors, have agreed to sponsor forums and workshops at INCOSE events to provide the best of commercial practice exposure to INCOSE Legacy Systems Engineers. The objective is to work towards a union of the best of both worlds. In addition, we recommend that INCOSE set a goal of 10,000 members, which would approximately triple the present membership. A majority of the new members would come from the commercial world.

An example of a strong INCOSE initiative towards the commercial world is the conference that the Colorado Front Range Chapter is hosting in Denver March 26–28, 2000. This conference is titled Systems Approach to Product Innovation and Development in Hyper-Competitive Environments. Don Clausing will be the keynote speaker and Technical Program/Paper Chair. Elliott Axelband and Sandy Campbell will work with Don to support this conference. You are encouraged to participate.

References

1 The Role of Concurrent Engineering in Weapons Systems Acquisition, IDA Report R–338, 1988.

2 The Cochairmen of the CETF were Dr. Don Clausing and Mr. D. Travis Engen, who is now the CEO of ITT Industries.

Biographies

Don Clausing *worked in industry for 29 years. Since then he has been at MIT for 13 years, where he is now the Xerox Fellow in Competitive Product Development in the Center for Innovation in Product Development.*

Elliott Axelband *worked in industry for 35 years. He retired from The Hughes Aircraft Company where he was a Group Vice President and Division General Manager. Since then he has been at USC for five years, where he is Associate Dean for Research Development, a Research Professor of Electrical Engineering, and Director of the Graduate Program in Systems Architecting and Engineering, and at RAND, where he is a resident consultant.*

Sandy Campbell *worked at Raytheon for 18 years, as a system architect/engineer and as vice-president for Research and Development, and was vice-president at Xerox for 17 years, managing technology development, technical strategy, and product development innovation. He has recently joined the Center for Innovation in Product Development at MIT.*

The Application of Systems Engineering to Forensic Investigations

Dr. Robert A. Warren, raw1@aol.com

Systems Engineering is a powerful approach to organizing and focusing complex technical, business and legal issues associated with disputes and incidents of failure.

Introduction. Systems engineering is an interdisciplinary approach to technical problem solving and, as such, is useful in organizing and disciplining forensic investigations and testimony. Systems engineering structures the forensic investigation as an iterative "problem, solution, verification" process. It helps to balance the case costs, schedule deadlines, and the broad technical, business and legal needs associated with a case and the dictates of the courts.

In this article, systems engineering definitions and concepts will be outlined, and then linked to the overall needs of the forensic investigation community.

What does a System mean to the Forensic Community. For the purposes here, a system is defined as an integrated composite of people, products and processes that provide a capability to solve a problem (or resolve a failure incident) under a specific set of circumstances[1].

A real or perceived incidence of failure drives the work of the forensic community. This failure incident, its associated environment, and legal practices constitute the basis for a system engineering examination. For example, computer software contract disputes are affected by the nature of a rapidly changing entrepreneurial industry and an equally rapidly changing litigation environment. On the other hand, ship operation accidents are affected by thousands of years of sea-going traditions and relatively stable national and international law associated with the safety of life at sea.

The problem part of the system definition provides the focus for incident investigation data gathering, testing, analysis, advice to the client attorney, court-required affidavits and reports, deposition and trial testimony, and an appeal of an adverse verdict. Basically, the system perspective helps to clarify incident causation and addresses the various probabilities and tradeoffs that support a successful case resolution for both an attorney and a disputant. And, success does not always mean a win, but rather the best resolution of the case under the circumstances!

To the forensic expert, the people, products, and process part of this definition links the litigants in a dispute with the instruments and methods that caused a failure. To ensure that a complete picture of the incident emerges, the "system" includes not only the incident and its relevant hardware, software, materials, data, techniques, facilities, and services, but also the legal process and its witnesses, experts, insurance companies, attorneys, laws, and courts.

The Systems Engineering Approach in Support of Forensics. Systems engineering is an interdisciplinary approach to evolve and verify an integrated and life-cycle balanced set of system product and process solutions that satisfy stated customer needs (and, therefore, help to resolve a failure incident).

Interdisciplinary thinking insures that an expert team and an attorney client have a common vision of the failure incident and case needs. It is not unusual for a systems engineer to be the investigative team leader to ensure that resources and efforts are effectively focused on understanding the problem, and eventually a judge and jury (or for that matter and insurance company). The SE ensures that the requirements of the legal process do not confuse, bias, or obscure the physical and functional realities of the incident.

Life cycle refers to the overall implications of a failure incident. In a boating accident, for example, there is a life cycle profile for the boat, for the individual who operated the boat, and for the way in which the boat was used. Referring to the boat itself, past considerations may involve issues of compliance with boat industry standards, and practices at the time of design and manufacture. Future considerations, such as the possibility of watercraft prohibition legislation, may affect the likelihood of case settlement. As a forensic approach, systems engineering provides a correlated analysis of life cycle product (boat) and process (design, manufacturing, and other) issues within legal context.

Balanced means that there must be a clear understanding and, ultimately, acceptance of the costs, schedules, needs and risks of the case. A good systems engineering effort balances case costs, litigation schedules, and "discovery" needs, and, ultimately, supports good decision making based on case risks, essentially the prospects for a successful settlement, trial outcome, or appeal. Poor case management effort can result in huge sums of money being lost in an attempt to compensate for case development mistakes or inadequacies, and can often lead to unsatisfactory outcomes that can pit client against attorney, and attorney against expert. In one memorable case, nearly a million dollars was fruitlessly spent to pursue the interests of a one-eyed individual with a very high blood alcohol level who, during the dark of night, hit an anchored lighted sailboat while driving his performance boat at high speed on a small lake. The attorney/expert relationship could not be maintained when the needs of the injured boater clashed with the truth of the situation. Eventually, a judge issued a summary judgment, throwing the case out of court.

Systems engineering provides for the development of simultaneous product and process investigations. In the forensic arena, this means that if a boat was involved in an accident, then there was a life cycle process involved. The elements of this process included:

- A boat designer;
- prototype development, testing and data analyses;
- the creation and adequate use of manufacturing processes and manufacturing standards, techniques and facilities;
- additional production model testing and documentation;
- sales and marketing efforts through marinas or other appropriate retail or wholesale outlets;
- some level of operational and safety training of users;
- appropriate paperwork to document the sales transactions and warranties;
- information and expert technical help to support, maintain and operate the boat throughout its useful life; and,
- disposal through public or private means.

Systems engineering, by helping to define the failure incident parameters and outline the scope of investigation, focuses limited attorney resources on the most productive aspects of case development and assures that relevant information is uncovered during the "discovery" process of the courts.

There is no need, for example, to

examine the design of the boat if it functioned properly in a waterway. The problem in such a situation may be an interaction with a wave, not the failure of the hull. On the other hand, if the hull delaminated and a boat passenger was injured or killed as a result, then such boat related issues as design, development, manufacturing practices, quality assurance, instructions and warnings, age, condition and wear life would be included in the accident examination.

There are many customers who are interested in a dispute. The victims, defendants, insurance companies, attorneys, judges, juries, and experts are closely involved in the legal process associated with a single case. In addition, there may be media, government, standards setting organizations, competitors, and the like who are interested in the broader implications of the situation. Systems engineering helps to maintain the focus on the objectives of "case" customers and to sort out the conflicts and confusion associated with external parties. This is done by establishing a clear problem definition and scope, creating an associated team charter, and using the definition and charter to sort risks and decisions into those which must be effectively controlled by the team and those which are outside the scope of team activities.

The Systems Engineering Process as a Working Tool for the Expert. The systems engineering process requires an input defined in terms of a problem (or, in the case of forensics, a failure incident) to be solved, and its associated context and constraints. What follows is a failure incident analysis which challenges the completeness, accuracy, and coherence of the incident description, begins to focus all members of the forensic team on the incident, and organizes the existing and needed facts and evidence into the analytical framework provided by systems engineering.

The importance of a good failure incident analysis cannot be underestimated. Forensic problem statements are initially structured by an attorney based on the recollections of a disputant and other fact witnesses, and often contain client and legal biases. In one instance, the attorneys, prior to the filing of a lawsuit, laid out the problem as they understood it and invested in a systems engineer to help structure the cost-benefit aspects of the case. When a failure incident analysis was conducted, the attorneys realized that they had identified the wrong problem, had invested up to that point in an incorrect manner, and were about to hire the wrong experts. They subsequently backed out of the case and thereby avoided the embarrassment of having to tell the client about the mistake.

The failure incident analysis is followed by a breakdown, or functional analysis and allocation, of what specific actions must be taken to understand the incident and its legal implications and how these actions must be allocated among expert team specialists and combinations of experts. By way of example, a boat accident forensic team might allocate responsibilities as follows:

- a systems/marine engineer to provide the boating perspective;
- a human factors expert to provide generalized safety advice;
- an economist to calculate losses;
- a doctor to articulate the severity of injuries and the long term prognosis;
- an instrumentation specialist to insure the integrity of data taking during testing and to work with a computer modeler; and
- a photographer to document evidence gathering and testing.

The results of diverse specialist activity must eventually be integrated to form a realistic and legally acceptable representation of case issues. Simply, the explanation, or more likely, alternative explanations of an accident or criminal incident, must conform to known and accepted technical, business, and legal practices and must be accurate and consistent. The gaps and overlaps that occur when integration is mishandled can, and almost always do, lead to conflicts in testimony and loss of credibility in the presentation of a case to a jury.

To ensure that the resolution does, in fact, solve the problem effectively, verification via testing, experimentation, modeling, simulation, inspection, and/or analysis is often done. This does not mean, however, that juries are required to accept the results of a verification process as proof. (Strange verdicts that defy logic, the laws of physics and common sense do occur. That is one of the reasons that some highly articulate attorneys often prefer trial to settlement.)

Problem defined, resolution(s) developed, and verification performed adds information to the legal "discovery" process, clarifies risks, and may lead to early case settlement. When the process is repeated, and the information output of a problem, resolution, verification cycle is incorporated into the evolving understanding of the incident, then the case becomes increasingly clear and the attributes of case resolution and case prospects better understood.

Control of the Systems Engineering Process and its Implications for the Expert. Configuration management is the systems analysis and control tool that is applied to the identification, documentation, and control of a product or process, including its basic form and any modifications, updates, and changes.

Configuration management is commonly divided into categories, namely the management of the situation itself, the management of the data that is relevant to the situation, and the management of the interfaces between the parties. Most investigations are initially and rightfully focused on the failure incident itself. Hopefully, care is taken to collect and preserve evidence that essentially defines the configuration of an item involved in a dispute. Complete evidence, however, is a rarity — particularly in accident situations where damage occurs. Subsequent test and analysis activities, when combined with evidence, constitute a configuration baseline and insure that nothing was knowingly done that violated incident boundaries and limitations. Frankly, failure to understand and control configuration is a prescription for disaster in forensic investigation and testimony, as testing, analysis and modeling is often done using an "exemplar," an item which is to the greatest extent possible like the item involved in the disputed incident. In one civil litigation, the plaintiff attorney did not know what propeller and skeg configuration was in evidence at the time of a bass boat "spin out" accident, but was claiming that the configuration, regardless of what is was, directly related to the accident. Because of this lack of knowledge, testing complexity and cost rose by a factor of four to cover all possibilities. This still left the attorney with a major problem: Some tested configurations were stable and would lead to a conclusion of operator error; while other tested configurations were not stable and would lead to a conclusion that modifications made to the skeg, in particular, were causally related to the accident.

Data management is increasingly important because diverse and complex data from many specialties must eventually be analyzed, integrated, and focused on the failure incident. Data development, security, and usage in the legal environment are expensive, predominantly because of the growing use of sophisticated graphics and animation. Data development and presentation, therefore, should be considered a specific disciplinary specialty and cost center. Notably, while there are accepted standards and recommended practices for criminal data management and presentation, this is not necessarily the case for the civil sector. Regardless, data development and management activities should be directly traceable to the failure incident analysis.

Interface management by manufacturers and dealers is often a weakness that is exploited in the legal environment. For example, inadequate manufacturer/dealer relationships may result from poorly written or improperly understood contract relationships and/or instructions. In one incident, the manufacturer of a boat purchased a pedestal seat from a supplier and mounted it in his product without adequate quality assurance checks. A horrific and deadly accident, directly traceable to this quality assurance failure by both the manufacturer and supplier, resulted in a multi-million dollar judgment against them. In the case management arena, the confusion, misunderstandings, and gaps in analysis that sometimes occur between forensic experts and attorneys can often be traced to role and responsibility ambiguities at the interface.

Risk management is a systems analysis and control tool that identifies, classifies, assesses and proposes approaches to handle the inherent risks in situation and case management. Good risk management disciplines decision making by identifying success or failure probabilities. A poor risk assessment, which often follows inadequate configuration management, causes confusion, explodes costs, and leads to uncertain outcomes. In the forensic world, risk management applies to the quantity and quality of evidence and analysis, and eventually to the reasonableness of case theories and alternatives. It also applies to the ebb and flow of case prospects as information is developed, analyzed and applied to solving the problem. Risk management is a major factor in case closure decisions, where settlement versus trial is the prime consideration. Good attorneys pay close attention to an expert's concern about risks and will settle a case or modify case presentation strategies and tactics accordingly. Less good attorneys "go shopping" for an expert that will tell the story the way they want it told.

Conclusions. As criminal and civil cases become more complex, attorneys who are not technical experts are confronted with the need for expert teams and, ultimately, a single point of contact within the expert team who can translate the issues of the technical community into an argument that can be used in the courts. The systems engineer is almost always that individual, because he or she brings the power and effectiveness of the systems engineering philosophy and framework to the case.

In general, the systems engineering philosophy strongly supports legal "discovery" activities, including deposition, trial information strategies and testimony, and, increasingly, appeal procedures. The systems engineering framework provides a practical problem solving mechanism that increases the probability of case success by providing discipline to evidence gathering and development, quantifying risks, and clarifying decision-making[2].

References:

Blanchard, B.S. and Fabrycky, W.J. (1990) Systems Engineering and Analysis, Second Edition. Prentice Hall International Series in Industrial and Systems Engineering, New Jersey: Prentice Hall, Englewood Cliffs, N.J. 07632.

Warren, R.A. (1997). The Effective Expert Witness: Proven Strategies for Successful Court Testimony, Chapter IV, Virginia Gaynor Publishing, P.O. Box 462, Lightfoot, VA 23090

Notes:

1 There are numerous definitions of "system." In the first reference, which is considered to be the seminal work in the field of systems engineering, a system is define variously as "an assemblage or combination of elements or parts forming a complex or unitary whole; any assemblage or set of correlated members; an ordered and comprehensive assemblage of facts, principle, or doctrines; a coordinated body of methods or a complex scheme or plan of procedure; and/or any regular or special method of plan or procedure." This author will use a version of these definitions that is commonly accepted throughout the systems engineering community.

2 The second reference provides more detailed insight into the use of the systems engineering process for forensic investigations, and gives practical "lessons learned" advice on roles, responsibilities, and problems faced by experts.

The Information ByWay

Challenger Revisited, Lessons for Systems Engineers?

Jack Fisher, seajnf@aol.com

It's been over 13 years now since the tragic loss of the Space Shuttle Challenger. This seems like an appropriate time to reexamine the circumstances of the accident and try to draw some conclusions that may help us in the practice of good systems engineering. I have no interest in assessing blame and I will not claim that better systems engineering would have prevented the loss. However, there are lessons to be learned. The definitive study of the Challenger accident was conducted by Diane Vaughan, a sociologist at Boston College, and reported in her book *The Challenger Launch Decision*.[1] I will summarize her findings, and then examine the circumstances to see what lessons there are to be learned by systems engineers.

The launch of Challenger occurred at 11:38 am on January 28, 1986 after overnight temperatures in the low twenties (degrees Farenheit). The ambient temperature at the time of launch was 36˚F. A leak at the aft field joint on the right Solid Rocket Booster (SRB) caused loss of the vehicle and crew at 73 seconds after liftoff. An escaping jet of hot gas caused the SRB to rotate and strike, in turn, the orbiter wing and the external tank, causing the tank to rupture and explode.

The SRBs provide the bulk of the thrust for the first 120 seconds of flight. The SRBs are shipped to the launch site in four segments plus the nozzle. Four field joints are required to mate the pieces of each SRB. There are eight joints to be sealed for the two SRBs. The gaps between the tang and clevis of the joining segments at each field joint are sealed with redundant circumferential Viton O-rings, 0.28 inches in diameter and about 37 feet in length. They are protected from the heat of combustion by an asbestos-filled zinc chromate putty. The O-rings are compressed upon installation. The internal pressure resulting from the SRB ignition transient causes a short-term joint rotation, which opens up the gap filled by the O-rings. The resiliency of the Viton should cause the gap to be filled. However, the impact of temperature on resiliency had not been fully characterized.

Prior to the loss of the Challenger, two types of anomalies were encountered with the O-rings, erosion and blow-by. Erosion of an O-ring by hot combustion gases was first experienced with the second shuttle flight in 1981. On the 24 flights prior to Challenger, erosion was experienced on 12 flights, including one flight where erosion occurred on the secondary O-ring.

The other anomaly, blow-by, occurs when the O-ring fails to seal the joint and allows hot gas to pass by. Blow-by was experienced on nine of the 24 flights prior to Challenger. The most serious case occurred in January of 1985 during the launch of STS 51-C when four joints experienced blow-by. Immediately prior to the launch, Florida had experienced several nights of record low temperatures ranging from 18 to 22˚F. Ambient temperature at launch was 66˚F. Later analysis indicated that the joint temperature was 53˚F, the lowest temperature experienced before Challenger. The correlation of blow-by with temperature was left uncertain, however, with the flight of STS 51-G later in 1985 when blow-by occurred at three joints despite a temperature of 70˚F. Altogether, blow-by was experienced at joint temperatures ranging from 53˚ to 76˚F.

Ms. Vaughan's analysis concludes that the "launch decision was rational calculation, but not amoral, that it was a mistake, but not misconduct." She goes on to describe the NASA/shuttle/SRB culture at the time and explains why it persisted despite the mounting evidence of problems with the SRB design.

The persistence was based upon what she refers to as the "culture of production" and "structural secrecy." She uses the terminology, "normalization of deviance" to describe the routine acceptance of problems with the SRB field joints and O-rings, despite accumulating evidence of problems during shuttle design, test and operational flights through 1985. Even as the problems worsened with the 1985 flights, this was considered acceptable. Richard Feynman, the Nobel Prize winning physicist, described it in Reference 2, "The argument that the same risk was flown before without failure is often accepted as an argument for the safety of accepting it again. Because of this, obvious weaknesses are accepted again and again, sometimes without a sufficiently serious attempt to remedy them, or to delay a flight because of their continued presence."

The NASA/shuttle/SRB project culture and the acceptance of risk, as described by Ms. Vaughan, were based upon a number of factors. The most important was the NASA desire to project that shuttle flights were safe and routine. The shuttle program included only four test flights and after the fourth flight in mid-1982, the shuttle was declared operational. Later shuttle crews included members of Congress, a Saudi Prince and, with Challenger, the Teacher In Space. The culture and experience of the SRB team included the beliefs that the O-rings were redundant, were an acceptable risk, any problems were self-limiting, and flight results were within predictions and experience. It was recognized that there had been problems at low temperature, but this was not a major concern because low temperatures were the exception in Florida. The "culture of production" refers to a situation where decisions are made on the basis of cost and schedule rather than technical or safety considerations, representative of a production environment. This creates pressure, spoken as well as unspoken, to continue production rather than delay to fix design problems, "If it ain't broke, don't fix it." The shuttle was sold on the basis that it would fly 60 times every year, carrying 65,000 pounds of payload each flight with a cost of $100 per pound of payload. By the early 1980s NASA's goal had shrunk to 24 shuttle launches per year to be achieved by 1990. There had been nine launches in 1985 and 15 were scheduled for 1986. By January of 1986 everyone involved was feeling the increasing pressure.

One of the responsibilities that we have to accept as systems engineers is that of being a technical conscience. We have to be willing to stand up and present the technical side of any issue to project management to counterbalance the cost and schedule considerations. For the situation where cost and schedule prevail, the technical/quality/safety risks must be clearly presented and made known to all concerned.

The issue of "structural secrecy," as defined by Ms. Vaughan, refers to the patterns of information access and availability within NASA and probably many other technical organizations. Much of the information created with the design of a system is very specialized and is available only in those areas of an organization with a direct concern. Even in the case where the data may be available, a specialist is required to interpret it and act upon it. As she also points out, where procedures require that information be widely distributed, there is likely to be so much information that it is routinely ignored. Further, at management levels, there is a "systematic censorship" based upon mechanisms designed to reduce information overload and call attention to certain selected information.

Here there are several systems engineering issues. Systems engineers have a collective responsibility to understand the complete system as well as all of its components, must serve as information/data brokers and know where to find data and who on a project needs to have access to it, and must serve as a spokesperson for the specialists on a project who might not otherwise be understood.

The issue of launch temperatures is critical in determination of the cause of the accident. The ambient temperature at launch was 36˚F and later analysis revealed that the SRB aft joint temperature was 31˚F. The SRBs were required to be qualified over a temperature range of 40˚ to 90˚F. The shuttle system requirement is to launch at any temperature between 31 and 99˚F. The system and its components should be qualified to a wider temperature range that required for launch. A typical qualification temperature margin is 10˚F so that the system and all of its components, including the SRB, should have been qualified over the range of 21˚ to 109˚F. SRB testing included motor firings at joint temperatures of 84˚, 49˚, 61˚, 40˚, 58˚ and 52˚F, and four qualification tests temperatures of 83˚, 67˚, 45˚ and 60˚F. There was no evidence of either O-ring erosion or blow-by during any of the demonstration or qualification motor tests.

The SRB qualification requirements are less severe than the system requirements. This represents a fundamental breakdown in the systems engineering process: A failure to properly allocate system-level requirements to the component level. One of the responsibilities of the systems engineer is to assure that all lower-level requirements are traceable and consistent with system requirements. In this role, a systems engineer will act as a surrogate for the user and assure that subsystem requirements are validated against system-level or user requirements.

The large thermal mass of the SRB results in temperature lag with respect to ambient air temperatures. This is evidenced by the difference, already noted, between the ambient temperature and the SRB aft joint temperature at the time of launch. Earlier on the morning of launch, a crew working on the pad took some measurements of SRB temperatures with an infrared pyrometer. Their readings, at 8:45 AM, indicated the left SRB (in direct sunlight) was 25˚F, and the right SRB (in shadow) was 8˚F, while the ambient temperature was 30˚F.

The launch criteria should be based upon a detailed thermal model of the system to ensure that all components are within their flight limits. The thermal model should provide a detailed representation of the system as well as solar orientation. Although systems engineers may not be directly involved in launch operations, they have a responsibility to ensure that the launch criteria and procedures will allow safe operation of the system. System design not only includes the hardware and software, but the data and procedures as well.

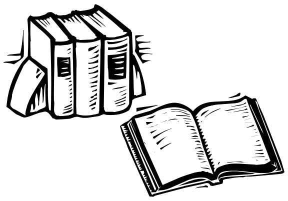
To summarize, the Challenger lessons learned are that systems engineers should:

1) serve as technical conscience for the system and be prepared to present the technical issues and risks as contrasted to cost/schedule concerns,
2) assure ready access to and availability of all project technical information,
3) ensure that lower-level component requirements are traceable and are consistent with system-level requirements
4) ensure the availability of mission procedures and constraints that allow safe system operation.

References:

1. *Diane Vaughan; "The Challenger Launch Decision, Risky Technology, Culture, and Deviance at NASA." University of Chicago Press, 1996.*

2. *Richard P. Feynman, "What Do You Care What Other People Think?" W. W. Norton, 1988.*



Book Reviews

Systems Architecting of Organizations: Why Eagles Can't Swim

by Eberhardt Rechtin, CRC Press, 1999, ISBN: 0849381401.

Reviewed by Lawrence D. Pohlmann, pohlmannld@erols.com

Pause and reflect, or, as bridge players would say, 'Review the bidding please.'

This is one of the insights that Rechtin includes in this, his third book on applying heuristics to the process of architecting. And pause and reflect is, indeed, what Dr. Rechtin has done. Citing numerous examples, anecdotes, and "caselets," he has distilled his experiences from a long and distinguished career in industry, academia, and government into 126 brief, memorable, and useful, insights. This time Rechtin's focus is on applying these insights or heuristics to organizations, to help them manage and cope with change. In the sections below, commentary by me is interspersed with selected direct quotes from the book.

Why Insights? We are in a living in sound bite society — we are bombarded with brief headline news stories, 15-30 second commercials, frequent and brief interruptions at the office, "factoids," and motivational quotes from famous people sprinkled on our walls and in our newspapers. Thus, we have become used to receiving news, information, guidance and advice in small, focused doses. Rechtin's packaging of knowledge, lessons learned, rules of thumb, and sage advice into concise, well-stated insights fits well in this environment. At the same time, Rechtin acknowledges the long heritage of the concept of insights: "…the Chinese recognized the value of insight in their familiar: A picture is worth a thousand words."

I feel that the utility of insights, which Rechtin defines as "perceptive understandings of the underlying nature of things," is not only in their concise statement of principle or knowledge, but in that they serve to facilitate contemplation, thinking, communication, and innovation — in individuals, within and among groups, and within and among larger organizations. Insights can serve to anchor a wide range of relevant and more detailed issues and guidance information. As he states: *"An insight is worth a thousand analyses."*

The Title Metaphor. The eagle has long been "A metaphor for excellence, strength, courage, and pride." Yet all "excellent organizations… know that eagles can't swim, regardless of motivation." Thus another insight: *"Given an excellent organization successful in its own field with objectives, skills, and policies designed for that success, there are some things it can not do — or at least not do well."*

Numerous examples are given throughout the book where eagles (i.e., excellent organizations) were less than successful in contexts requiring organizational change, evolution, or agility.

The Author. Rechtin has held key senior executive positions in industry and government, and has received numerous awards, including INCOSE's own Pioneer award. In his own words he "has had the rare privilege of working as an executive in six truly excellent organizations." He is a respected educator. He previously authored *Systems Architecting, Creating and Building Complex Systems,* 1991, and (with Maier) *The Art of Systems Architecting* (1997). He truly has the credential for authoring the current book. And just as important, he writes extremely well. To steal a phrase from reviewers of more popular books, Rechtin "is a good read!"

The Target Audience. "This book is primarily written for professionals and managers who are in excellent organizations…faced with the prospect of unexpected change." In a broader sense, I view the book as useful for anyone who must deal with or appreciate the opportunities and challenges of organizational evolution and change. Books like Rechtin's can help engineers, professionals, managers, and executives cope with, and help to evolve and improve, the organizational environment in which the project and product architecting, engineering, and design must be performed.

The Motivation. "The book was written because highly respected organizations…are now confronted by a very difficult dilemma…the demands of excellence on one hand and of change on the other can be cruelly irreconcilable." Rechtin is convinced, and I agree, that many of the principles or insights of architecting that work in product design and development can also be effectively applied to organizational change management. Rechtin's recommended "approach is architectural and heuristic," rather than strictly scientific and heavily quantitative.

The Premises. The book has four basic premises:

1. Organizations are complex systems…
2. Every system and organization has an architecture…
3. Systems architecting can be… applicable to the structural problems of organizations…
4. Systems architecting insights… can be effectively used to sustain the excellence of organizations…

This is not to say that organizations can be architected, engineered, and designed in the same way that products are architected, engineered, and designed, but rather that some of the same principles can and do apply.

The Structure. The book is itself well architected, engineered, and designed to serve multiple purposes and multiple types of users, i.e., text, primer, guidebook, resource and reference. It is as if Rechtin was aspiring to one of the book's stated insights: *"It is a beautiful thing when it is all working together."* To me, this book is a beautiful thing! The pieces work together extraordinarily well.

The book's five parts, with a total of 11 chapters, each have a different function and provide a different perspective on organizations and the applicability of the tools and insights of architecting. Rechtin states "The central purpose of the book is built into the book's structure." Readers are encouraged and assisted to view organizations as complex systems (Chapter 1), as creators of emergent values (Chapter 2), as competitors (Chapter 3), as partners with government (Chapter 4), as sets of beliefs (Chapter 5), as structures (Chapter 6), and as sets of interlocking decisions (Chapter 7). These seven chapters tend to be analytic and descriptive; they are really there to provide the background and context for Chapters 8 through 11. The later chapters are more prescriptive in addressing the opportunities and challenges in the architecting of organizations. Each chapter contains a logical sequence of chunks of relevant information, the chunks themselves being of a manageable and useable size. Esoteric jargon is notably absent.

The chapters are complemented by appropriate introductory materials, a detailed table of contents, clear definitions of all key terms (presented early in the book), and useful, well-organized appendices. From the elegance of the structure and the extent of cross-referencing, one may infer that Rechtin had internalized a detailed concept of how his book could and would be used — and by what kinds of people. Indeed form does follow function. One could even reasonably conclude that the book itself was consciously architected.

In discussing the structure, Rechtin observes, "All these perspectives are interlinked, none are quantitative, and none are either right or wrong. All suggest approaches but none are mandates. And just like architecting itself, they take a user's wisdom to bring them to life."

A Sampling of Insights. It is highly likely that readers will identify with a number of the insights presented, and will easily resonate with many more. Each insight is discussed in the text, and supporting evidence or anecdotes are presented. Several insights are discussed in multiple contexts. The 126 insights, grouped into nine categories, are again listed in Appendix B, with keywords bolded, thus making it easier to locate the insights that may be of interest at the moment. Among the insights listed are the following:

- "No system can survive without a viable purpose."
- "Relationships among the elements are what give an organization its added value."
- "Don't assume that the original statement of the problem is necessarily the best, or even the right one."
- "In open competition the incumbent has the encumbrances."
- "The most dangerous assumptions are the unstated ones."
- "All the serious mistakes are made in the first day."
- "To be successful requires a diversity of perspective, experience, education, and belief."
- "When integrating two organizations, distinguish between the real synergies and the perceived ones early, and promptly activate teams that can demonstrate that reality."

Many of the examples, stories, anecdotes, and caselets that are included in the discussion of the insights provide an interesting and fascinating commentary of the interrelationships and interdependencies of product development and organizational development issues.

Availability. The book is available in bookstores, directly from CRC Press (www.crcpress.com), from Amazon.com (which also includes reviews of Rechtin's earlier books, including a review by Rechtin himself), and from technical book outlets (e.g., San Diego Technical Books, at www.sdtb.com). The book was published as part of CRC Press's Systems Engineering Series.

In Conclusion. The INCOSE community should easily identify with Rechtin's approach, his examples and anecdotes, and his conclusions. He understands our discipline. He thinks and reasons about organizations and organizational change and evolution in terms that are familiar to the systems engineering practitioner. I would particularly recommend the book to those of us involved in any way with organizational or process change.

Lastly, perhaps INCOSE should have an insight of our own:
If the book is by Rechtin, it's relevant, it's a good read, it's a good resource, and it's likely to be reusable in a number of contexts!

Working Knowledge: How Organizations Manage What They Know

Davenport and Prusak, Harvard Business School Press, ISBN: 0875846556

Reviewed by Virginia Lentz, virginia.lentz@otis.com

[*Reviewer's Note:* I heard Prusak talk and ordered the book! If I had read the book before hearing him, I would have asked whether the title is verb:noun or adjective:noun! I will use random extracts of the book out of order to present the review. There is fodder here for the process wonks, the folks driving the strategic direction of INCOSE, and the intuitive System Engineer. We are knowledge workers!]

A company's ability to produce is embedded in the routines and machinery of production. The material assets of the firm are of limited worth unless people know what to do with them. Knowledge markets don't operate very efficiently: Unrecognized, disorganized, local, and often discouraged rather than fostered by company culture, these markets are deeply imperfect mechanisms for generating and exchanging insights. Knowledge born of experience recognizes familiar patterns and can make connections between what is happening now and what happened then. The expendable middle managers were key knowledge organizers and synthesizers, and downsizing companies mistakenly assumed that technology could replace the skill and judgment of an experienced human worker. Technology can enhance knowledge work, but only once we understand how knowledge is developed and shared. Knowledge brokers know where to go for knowledge, especially when it falls outside their area of responsibility.

Knowledge develops over time and includes what we absorb from courses, books, mentors, as well as informal learning. Experience will change ideas about what should happen, into knowledge about what does happen, and provides a grounded truth about what really works and what doesn't. Knowledge works through flexibility, and guides one to actions that are developed through trial and error, and over long experience and operation. Knowledge offers speed. Intuition offers compressed expertise. When knowledge stops evolving, it turns into opinion or dogma.

Knowledge is neither data nor information, and the three are not interchangeable concepts.

Understanding what these three are and what you can do with each is essential to doing knowledge work (or Systems Engineering) successfully. Knowledge derives from minds at work. It originates and is applied in the minds of knowers. In organizations, it often becomes embedded not only in documents or repositories but also in organizational routines, processes, practices and norms. Tacit, complex, knowledge that is developed and internalized by the knower over a long period of time, is almost impossible to reproduce in a document or database. Such knowledge incorporates so much accrued and embedded learning that the rules may be impossible to separate it from how an individual acts. In practice, it is difficult to locate the dividing line between knowledge that is fully embedded in a process and the tacit human knowledge that keeps the process going.

*And the reality is…*Without trust, knowledge initiatives will fail regardless of the effort involved. Without trust, knowledge initiatives will fail regardless of how thoroughly they are supported by technology and rhetoric and even if survival of the organization depends on success of the knowledge transfer.

People on the Move

Jim Haney, late of the Colorado Front Range Chapter and its 1997-98 president, re-located during late March 1999 from Colorado Springs to St. Petersburg, FL. He is still with Raytheon Systems Company - C3I, and has been assigned as the Senior Manager for Cooperative Engagement Capability Engineering Programs, which covers both systems and software development. Jim can be reached at 727-302-7956, or j.h.haney@ieee.org.

William H. McCumber, Jr., Ph.D., P.E., formerly of Lockheed Martin, took the position of President of EagleRidge Technologies Inc., in February 1999. As president of this consulting firm, he continues his work in systems engineering, course development, and training. In addition, Bill has been named a professor at the University of Maryland University College, teaching systems engineering at the graduate level over the Internet. His new address is EagleRidge Technologies, Inc., 118 Ledgerwood Lane, Rockwood, TN 37854. You can also reach Bill at 423-354-1500 or 423-354-3116, or mccumber@aol.com.

Thomas Nagle retired from his position within Boeing Long Beach, CA as Director of System Engineering to take the position of Chief System Engineer at the FAA. He can be reached at 202-493-4395 or by e-mail at thomas.nagle@faa.gov or corvette@pe.net.

George Percivall has joined SGT, Inc. in Greenbelt, MD. Previously George was with Raytheon in Landover, MD. Mr. Percivall brings his expertise in systems engineering, software development, and standards development for geographic data, in particular from remote sensing sources, to SGT and NASA projects. You can contact him at george@sgt-inc.com.

Mark Schaeffer has moved from Systems Engineering Directorate in the Office of the Under Secretary of Defense (Acquisition & Technology) to the Deputy Director for Management at the Defense Advanced Research Projects Agency (DARPA). He is very interested in Systems Engineering and the roll that INCOSE will play in the future. Mark can be reached at mschaeffer@darpa.mil.

Systems Engineering: The Journal of The International Council on Systems Engineering

Call for Papers

The *Systems Engineering* journal is intended to be a primary source of multidisciplinary information for the system engineering and management of products and services, and processes of all types. System engineering activities involve the technologies and system management approaches needed for:

- definition of systems, including identification of user requirements and technological specifications;
- development of systems, including conceptual architectures, tradeoff of design concepts, configuration management during system development, integration of new systems with legacy systems, integrated product and process development; and
- deployment of systems, including operational test and evaluation, maintenance over an extended lifecycle, and reengineering.

The *Systems Engineering* journal is the archival journal of, and exists to serve the following objectives of, the International Council on Systems Engineering (INCOSE).

- To provide a focal point for dissemination of systems engineering knowledge.
- To promote collaboration in systems engineering education and research.
- To encourage and assure establishment of professional standards for integrity in the practice of systems engineering.
- To improve the professional status of all those engaged in the practice of systems engineering.
- To encourage governmental and industrial support for research and educational programs that will improve the systems engineering process and its practice.

The Journal supports these goals by providing a continuing, respected publication of peer-reviewed results from research and development in the area of systems engineering. Systems engineering is defined broadly in this context as an interdisciplinary approach and means to enable the realization of successful systems that are of high quality, cost-effective, and trustworthy in meeting customer requirements.

The *Systems Engineering* journal is dedicated to all aspects of the engineering of systems: technical, management, economic, and social. It focuses on the life cycle processes needed to create trustworthy and high quality systems. It will also emphasize the systems management efforts needed to define, develop, and deploy trustworthy and high quality processes for the production of systems. Within this, *Systems Engineering* is especially concerned with evaluation of the efficiency and effectiveness of systems management, technical direction, and integration of systems. *Systems Engineering* is also very concerned with the engineering of systems that support sustainable development. Modern systems, including both products and services, are often very knowledge intensive, and are found in both the public and private sectors. The Journal emphasizes strategic and program management of these, and the information and knowledge base for knowledge principles, knowledge practices, and knowledge perspectives for the engineering of systems. Definitive case studies involving systems engineering practice are especially welcome.

The Journal is a primary source of information for the systems engineering of products and services that are generally large in scale, scope, and complexity. *Systems Engineering* will be especially concerned with process or product line related efforts needed to produce products that are trustworthy and of high quality, and which are cost effective in meeting user needs. A major component of this is system cost and operational effectiveness determination, and the development of processes that assure products that are cost effective. This requires the integration of a number of engineering disciplines necessary for the definition, development, and deployment of complex systems. It also requires attention to the lifecycle process used to produce systems, and the integration of systems, including legacy systems, at various architectural levels. In addition, appropriate systems management of information and knowledge across technologies, organizations, and environments is also needed to insure a sustainable world.

The Journal will accept and review submissions in English from any author, in any global locality, whether or not the author is an INCOSE member. A body of international peers will review all submissions, with potential author revisions as recommended by reviewers, with the intent to achieve published papers that:

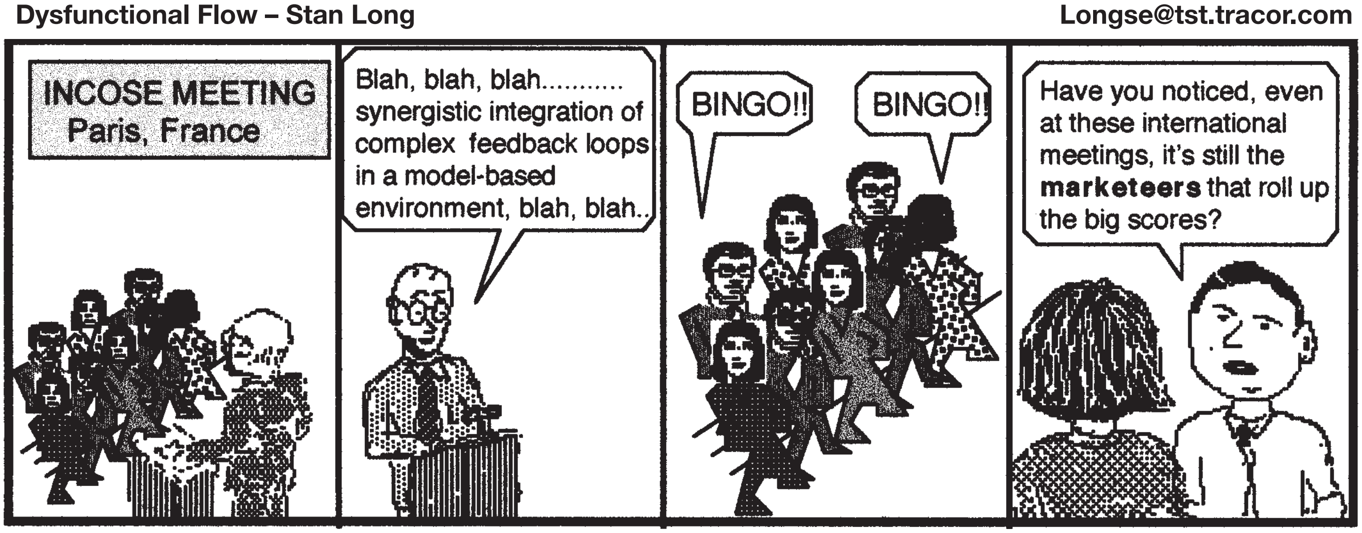
- Relate to the field of systems engineering
- Represent new, previously unpublished work
- Advance the state of knowledge of the field
- Conform to a high standard of scholarly presentation

Editorial selection of works for publication will be made based on content, without regard to the stature of the authors. Selections will include a wide variety of international works, recognizing and supporting the essential breadth and universality of the field. Final selection of papers for publication, and the form of publication, shall rest with the Editor.

Submission of quality papers for review is strongly encouraged. The review process is estimated to take three to five months. Five copies of your manuscript should be submitted for review purposes to:

Professor Andrew P. Sage
Editor in Chief, *Systems Engineering*
School of Information Technology and Engineering
George Mason University
Fairfax, VA 22039-4444
TEL: 703-993-1506 FAX: 703-978-9716
EMail: asage@gmu.edu

Dysfunctional Flow – Stan Long **Longse@tst.tracor.com**



Do you have ideas for Stan's next cartoon? Contact him at longse@aol.com

2000 SYMPOSIUM CALL FOR PAPERS

"Systems Engineering: A Decade of Progress and A New Century of Opportunity"

Tenth Annual International Symposium of
The International Council on Systems Engineering

Hosted by the North Star and Heartland Chapters

Minneapolis, Minnesota • July 16–20, 2000

The symposium theme is Systems Engineering: A Decade of Progress and a New Century of Opportunity. The 2000 Symposium will look back at the first ten years of INCOSE and its growth to today's international organization with more than 30 local chapters around the world. And the symposium will look forward to a new century and the opportunities for systems engineering to continue its increasing role in worldwide progress and growth.

Papers are invited which can make a contribution to the overall theme of the Symposium. All types of paper will be considered, from case studies to development work to technical analysis. Clarity of the message and its effective communication will be scored as highly as content. Papers must be submitted in English, the official language of the INCOSE Symposium.

Technical Tracks:

- SE Applications
- SE Management
- Education and Training
- Measurement
- Modeling and Tools
- Processes and Methods

Areas of Special Interest are:

- development & use of standards
- business process applications
- metrics and assessment
- model driven design
- testing & evaluation techniques
- tools integration & database use
- use of information technology
- requirements management
- collaborative working
- management of complexity
- reuse / use of standard parts
- international factors
- large scale systems
- risk management
- people management
- professional development
- small project applications
- system design techniques
- software engineering
- integrating systems
- use of formal methods

Draft papers of between 4 and 8 pages (including graphics) shall be submitted in electronic form, via ftp, e-mail, or disk. For format and address details refer to the INCOSE website: http://www.incose.org, where details associated with this Call for Papers will be posted. Papers will be reviewed anonymously—author information is provided in a separate file. Authors will be given reviewers' comments. Successful authors may also be contacted by Session Chairs with further comments on their papers, intended to fit them with the other successful papers into the developing session and track themes. Joint authors must nominate a single point of contact. Papers will be published in the symposium proceedings.

TECHNICAL QUESTIONS:	SEND PAPERS TO:	IMPORTANT DATES:
Pradip Kar United Defense LP 4800 East River Road Minneapolis, MN 55421 USA Tel: 612-572-4722 Fax: 612-572-4912 E-mail: pradip_kar@udlp.com	**Cass Jones** Professional Conference Management, Inc. 7916 Convoy Court San Diego, CA 92111 USA Tel: 619-565-9921 Fax: 619-565-9954 E-mail: pcminc@pcmisandiego.com	**Draft Paper Submission:** November 1, 1999 Acceptance Notifications: February 7, 2000 **Final Papers Due:** April 3, 2000

Minneapolis 2000

Call for Exhibitors

Pick up

Spring 99 issue

Membership
App

What's Inside

Editor's Note 3

President's Corner 4

***INSIGHT* Special Feature:**

Seven Insights into Process Modeling and Management for IPPD

Modeling Concurrent Engineering Processes in an Integrated Product Development Environment 6

Modeling Integrated Product, Prospect and Enterprise Development 11

Process Modeling with Design Structure Matrices 15

A Model-Based SEMP Approach 18

The Application of System Dynamics to Concurrent Engineering 20

The Common Denominator 24

Development Integration and Implementation of Engineering Processes at Oerlikon Processes 29

Working Groups

Tools Integration and Interoperability WG 34

The MWG Helps Practical Software Measurement Gets A Systems Perspective 34

Measurement: Frequently Asked Questions 36

Liaison between INCOSE and ISO 10303-AP233 Working Group 38

SE Applications Technical Committee Activities 39

News from Chapters 41

INCOSE Infrastructure

Volunteering As You Like It 47

New Ways & Means Co-Chairs 47

Call for Nominations for INCOSE Fellows 48

The Power of One 36

Commentary

Advances In Commercial Product Development Lessons for INCOSE SE 50

The Application of SE to Forensic Investigations 53

The Information ByWay

Challenger Revisited, Lessons for Systems Engineers 57

Book Reviews 59

People On The Move 61

Dysfunctional Flow/ Stan Long cartoon 62

INSIGHT

International Council on Systems Engineering

2150 N. 107th St., Suite 205

Seattle, WA 98133-9009

First Class Mail
U.S. Postage
PAID
Seattle, WA
Permit #3583